The Dissertation Committee for Matthew Thomas Harger Certifies that this is the approved version of the following Dissertation.

Development and Analysis of Tinker-OpenMM as a GPU-based Free Energy Perturbation Engine

Committee:

Pengyu Ren, Supervisor

Kevin Dalby, Co-Supervisor

Ron Elber

Walter Fast

Karen Vasquez

Development and Analysis of Tinker-OpenMM as a GPU-based Free Energy Perturbation Engine

by

Matthew Thomas Harger

Dissertation

Presented to the Faculty of the Graduate School

of The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2019

Acknowledgments

I want to thank Pengyu Ren and Kevin Dalby, and the members of both labs, as well as the rest of my committee for aiding in my development as a computational chemist.

I would especially like to thank Ju-Hyeon Lee, Ramakrishna Edupuganti, and Juliana Taliaferro for their synthetic and assay work that enabled my studies with MELK.

I want to thank Peter Eastman for his assistance in learning the OpenMM codebase. His help was invaluable in the early stages of adding free energy perturbation to Tinker-OpenMM

Finally, I would like to thank the Jay Ponder lab for useful discussion and assistance in the development of Tinker and Tinker-OpenMM

Abstract

Development and Analysis of Tinker-OpenMM as a GPU-based Free Energy Perturbation Engine

Matthew Thomas Harger, PhD

The University of Texas at Austin, 2019

Supervisor: Pengyu Ren and Kevin Dalby

The utilization of computational technologies for the lead optimization process is one of the biggest challenges in the computational chemistry field. In this dissertation, I describe the addition of GPU-based absolute and relative free energy calculation methods using polarizable force field AMOEBA to Tinker-OpenMM. I then proceed to test the capabilities of this platform by studying the binding free energy and binding structures of derivatives of the MELK inhibitor IN17. Also, I present the implementation of virial-based pressure control to the Tinker-OpenMM platform that is needed for performing isobaric simulations.

Table of Contents

INTRODUCTION **1**

Docking 2

Machine Learning 4

Quantum Mechanics 8

Molecular Dynamics 10

Free energy from molecular dynamics simulations 12

Forcefields 14

GPU Computing 17

Dissertation Overview 19

TINKER-OPENMM: ABSOLUTE AND RELATIVE ALCHEMICAL FREE ENERGIES USING AMOEBA ON GPUS **21**

Introductory Statements 21

Abstract 22

Introduction 23

IMPLEMENTATION DETAILS 29

Tinker-OpenMM interface 29

Absolute binding free energy 29

Dual-topology relative free energy 32

Methods 34

Simulation setup 34

Molecular dynamics 34

Bennett acceptance ratio 35
Hydration of aromatic compounds 35
Sampl4 host-guest binding simulations 36
RESULTS 37
Force Agreement 37
Computational efficiency 37
GPU/CPU absolute free energy agreement 39
GPU/CPU relative free energy agreement 39
Discussion and Conclusions 40
Concluding remarks 41
Tables 42
Figures 46
COMPUTATIONAL INSIGHTS INTO THE BINDING OF IN17 INHIBITORS TO MELK 52
Abstract 52
Introductory Statements 52
Introduction 55
Methods: 57
Parameterization 57
Simulation parameters 58
Complex structure generation: 58
Binding free energy simulations 59
IN17 Solvent Phase Crystal Structure 60

Results/Discussion 61

Aryl-Carbonyl Isomerism 61

MELK-nintedanib complex structural prediction 61

Absolute binding free energy of MELK with IN17 63

IN17 binding mode 64

Relative binding free energy of IN17 derivatives 65

The n-terminal loop structure is altered by substitution on the benzene (G2) offshoot 66

Effects of substitution on binding mode 67

Use of restrained equilibration to improve prediction 68

Entropy-enthalpy compensation 69

Conclusions 71

Acknowledgments 72

Concluding Statements 72

Tables 76

Figures 78

VIRIAL BASED BERENDSEN BAROSTAT ON GPUS USING AMOEBA IN TINKER-OPENMM 86

Introductory Statements 86

Introduction: 86

Methods: 90

Derivation of virial: 90

Virial Implementation: 91

Berendsen Barostat Implementation: 92

Virial Value Confirmation: 93

MD Procedures: 94

Molecular Systems: 94

Results: 95

Virial CPU vs. GPU Comparison 95

Equilibration of Small Molecule Systems: 96

Equilibration of Water at High Pressures 96

Comparison of Berendsen and Monte Carlo Barostats on GPU 97

Conclusions: 98

Concluding Remarks 99

Acknowledgments: 100

Tables: 101

Figures 104

FUTURE DIRECTIONS **106**

APPENDIX **110**

REFERENCES **114**

INTRODUCTION

One of the principal goals of the computational chemistry field has been the prediction of the binding affinity of small molecules to proteins[1]. Such capabilities would allow scientists to make actionable decisions in drug design before undergoing chemical synthesis and experimental testing. This would accelerate drug discovery, limit waste, and allow for a more significant structural understanding early in the drug optimization process. Current techniques are, however, often not accurate enough to achieve sub-kcal/mol accuracy in binding affinity prediction consistently[2]. This thesis chronicles my contributions to the efforts to achieve accurate protein-ligand binding predictions using the AMOEBA forcefield on GPUs, combining accuracy and speed.

High-level ab initio quantum mechanics (QM) calculations in principle could provide the ideal solution to the protein-ligand binding problem. However, given the prohibitive computational cost of quantum mechanics calculations on large systems of hundreds and thousands of atoms, computational chemistry is currently separated into fundamentally distinct techniques for different computational problems[3]. While pure quantum chemical calculations inform many of these approaches, approximations of molecular interactions are necessary for computational efficiency. The extent of these approximations defines the various subfields of computational chemistry. In this introduction, I will include a broad overview of the computational field as a whole, including docking, QM methods, machine learning methods, and an emphasized section on classical molecular mechanics-

based molecular dynamics (MD) approaches. This will be followed by an overview of GPU computation and the Tinker-OpenMM package for molecular dynamics simulations on GPU.

Docking

The basics of docking consist of a basic scoring function and a search algorithm for predicting optimal protein-ligand binding pose and affinity. Molecular docking is an approach that has been designed to identify possible small molecule binders of a target structural site on a protein or other macromolecule such as DNA [4]. Docking software requires only the input 3d structure of the target receptor, and a 3d structure of the ligand to be docked. The docking approach then attempts to find the lowest energy protein-ligand pose for a given protein and ligand pair. This pose is then assigned a score based on the predicted interaction strength[5]. The approach and scoring functions vary depending on the computational approach. The main two approaches used to generate low energy poses are molecular mechanics-based minimization using a relatively inexpensive and versatile forcefield (as used by GLIDE[6]) and genetic algorithms(like used in GOLD[7]).

The scoring function of a docking approach constitutes a simpler model of ligand-host interactions than even the most basic classical mechanics forcefields. Since bonded interactions (other than torsions) are assumed to be virtually identical across poses, bonded terms are excluded from scoring. Also, in order to improve upon computational throughput, scoring functions often use a simplified model of electrostatics. Instead of

calculating actual interaction energy, electrostatics interactions are often treated as hydrogen bonds with a score function related to distance and angle. This change in electrostatics is a necessary approximation, as most docking approaches do not take account of the solvent effect explicitly (though most packages allow for the utilization of crystallographic waters[8]). These crude approximations allow for efficient throughput, though this comes at a distinct cost of accuracy.

As a whole, docking approaches are successful at what they have been designed to do - namely, identify (enrich) possible ligand hits in large libraries of compounds consisting of millions of compounds[9]. However, docking approaches are ineffective in later-stage lead optimization, where chemical (sub kcal/mol) accuracy is necessary. Even the most accurate docking approaches struggle to obtain significantly better than a 2.5kcal/mol accuracy in binding free energy prediction[10-11]. The low accuracy of docking approaches can be attributed to several factors, most prominently the crude electrostatic model, entropy calculation, solved effect, and lack of system dynamics and induced fit effects. Also, most docking approaches assume that the host and ligand are static entities. In reality, both molecules are flexible, and the protein adjusts in pose due to the presence of a ligand, an adjustment known as induced fit. Also, it is the distribution of interaction energies that results in the experimentally observed potency, not the energy of an individual pose. Most docking packages enable the inclusion of sidechain torsional degrees of freedom in the pose optimization process[12]. This does capture some induced-fit effects, though it ignores basic backbone motions and secondary structural changes required during ligand binding. Approaches have been developed that allow for some

backbone flexibility, though this comes at a non-insignificant increase in system degrees of freedom, reducing performance[13-14]. Therefore, for docking to be most effective, the input protein pose must be in a state conducive to ligand binding. Ideally, this would consist of a co-crystal structure of the protein target with a related ligand in the binding pocket. The presence of a ligand allows for protein target to be in a more "relevant" state relative to an apo-protein, and thus more likely to be accommodating of a bound ligand. When utilized correctly, docking is a useful lead discovery tool. However, in later stages of drug discovery where higher accuracy is required, docking is mostly ineffective.

Machine Learning

Another increasingly popular class of computational chemistry methodology is machine learning approaches[15]. While the other classes of discussed techniques utilize the 3d structure of ligands, most machine learning approaches treat ligands as 2d atomic structures. While there has been increasing interest in the utilization of machine learning combined with 3d structural information to enable dynamics simulations[16], these approaches will not be discussed here. In this section, I will discuss the basics of machine learning, give an overview of its applications in computational chemistry, and discuss the advantages and disadvantages of these approaches.

Machine learning approaches often treat ligand structures like strings, such as SMILES strings[17]. For example, benzene in SMILES format is c1ccccc1, with the lowercase c representing an aromatic carbon atom, and the 1 indicating ring connections. The advantage of treating molecules as simple strings is the ability to gain access to the

diverse set of string analysis algorithms while ignoring a large number of 3-dimensional degrees of freedom. This utilization of atomic topology (atomic type plus connections) alone can result in a simplification of the amount of data needed in an analysis. For example, molecular toxicity is a combination of the interaction of ligands with many off-target proteins. In theory, one could design a 3d structural approach to predict toxicity based upon predicted binding to these off-target proteins. However, this approach would be computationally inefficient and require accurate binding predictions for each of the off-target proteins. In contrast, a machine learning model input would merely require an input library of compounds with associated properties. Using this mapping of input molecules to output properties, machine learning approaches attempt to generate a model that takes molecular features (such as topology, or other, provided precomputed metrics) as inputs and predicts the properties of interest.

Once a useful machine learning model has been generated, predictions could be made orders of magnitude faster than the binding free energy-based approaches. However, as the name implies, machine learning requires a sufficient training set of material inputs in order to generate a predictive model. For example, in the case of molecular toxicity prediction, this training set would consist of a series of molecules with known toxicity data. The learning method then attempts to create a series of equation constants that results in a predictive model that most closely reproduces the input training data. Without a large input training set, a useful (predictive) machine learning model cannot be generated. This creates problems for chemical property prediction, where obtaining a well-curated, sufficiently comprehensive dataset can prove challenging.

Despite these limitations, machine learning models have proven to be valuable in the drug discovery process and have permeated throughout the field. The most apparent utilization of machine learning schemes is in the prediction of ADME-Tox properties[18]. Since these properties are often independent of the protein target, a single, well-trained model can be used for aiding the design of ligands to any number of independent protein targets. Machine learning techniques have been used to develop models of excretion[19], distribution[20], drug-drug interactions [21], and more. Also, machine learning methodologies have been used successfully in the prediction of protein-ligand binding affinity[22-23], and perform well in protein-ligand affinity prediction tests[2]. However, this performance is enabled by the presence of the preexisting large amount of protein-ligand binding data, something most often lacking in new drug discovery projects. The most fruitful utilization of machine learning approaches in drug discovery has been quantitative structure-activity relationship (QSAR) models that use existing data to predicting binding or bioactivity towards already studied classes of proteins. For example, machine learning methods have been used to predict antifungal activity[24], as well as the affinity of inhibitors to HIV-1 protease, trypsin, and carbonic anhydrase[25]. These studies were likely successful due to the depth of data available to these extensively studied targets. The later study was also able to pick out features from a diverse set of input proteins and predicted the binding affinity of a test set of random protein-ligand complexes to within 1.6 kcal/mol. However, this approach was not able to reliably outperform traditional docking methods such as SYBYL:: ChemScore. While it is possible (if not likely) that machine learning methods will be utilized to inform and

improve traditional docking approaches[22]; machine learning is unlikely to obtain the accuracy needed for lead optimization studies.

The main advantage (as well as one of the most significant weaknesses) of machine learning approaches to molecular property prediction is the lack of a need for a solid mechanistic understanding of the process involved. Given enough input data, machine learning approaches will generate a predictive model, even in cases where the exact mechanism is not solidly understood. This is especially important in drug discovery, where most properties of interest are a combination of a myriad of factors, some of which may not be understood well enough to program into a predictive model directly. However, this advantage also means that it can be difficult, if not impossible, to interrogate a machine learning model to gain mechanistic understanding. A traditionally physics-based approach, at least, provides the ability to investigate mathematical and process intermediates for trends. For example, when calculating binding free energy using a molecular dynamics (MD) based approach, one could identify that a given electrostatic interaction is more negative in stronger binding compounds than weaker binding compounds, and thus possibly crucial in the mechanism of differential binding; or a binding site residue plays a central role in affinity or selectivity. However, machine learning approaches give no such insights. A trained machine learning model merely consists of a series of mathematical constants, each of which does not correspond to an identifiable physical phenomenon. It is also dangerous to extrapolate an ML model into unknown (or untrained) space, similar to mathematical spline functions that are designed to reproduce any complex surface given enough grid points. Therefore, while machine

learning approaches may be useful tools, deriving a better biophysical understanding using most machine learning processes is a largely non-viable process.

Quantum Mechanics

As the name implies, Quantum mechanics calculations depend on the calculation of the energy and wavefunction using the Schrödinger equation[26]

$$\left(\frac{-\bar{h}}{2m}\nabla^2 + V(r)\right)\psi(r) = E\psi(r)$$

These equations allow for analytical calculation of energy for simple systems, like the hydrogen atom or helium[26]. However, approximations need to be made to enable the approximate solution to energy for any larger molecular system. One of the "cheapest" methods is Hartree-Fock, the basis for most QM calculation methods[27]. One of the major assumptions made by Hartree-Fock is that electron motions are not correlated. This lack of electron correlation leads to some issues in forcefield development, mainly due to the lack of London dispersion, an interaction critical to the description of van der Waals (vdW) forces. Therefore, more expensive methods that make fewer assumptions have been developed, such as MP2[28] and CCSD(T)[29]. Also, the accuracy is defined by the basis set used, which describes the number of (often gaussian) functions that describe each electron distribution. The methodology and basis set size both contribute massively to overall computational cost. This is critical, as quantum calculations scale with the number of electrons on the order of $O(N^4)$ for standard methods like Hartree-Fock to $O(N^7)$ for expensive methods like MP4 [30], meaning that a doubling in the number of

electrons results in at least a 16 fold decrease in performance. This lack of scalability means that QM methodologies are limited in overall system size, making the most expensive methods inaccessible for biopolymers such as proteins or nucleic acids.

Despite limitations in the performance of QM methodologies, QM has proven essential to the computational chemistry community[31] given the "ab initio" nature. QM tools constitute the most definitive approach to the determination of the strength of individual interactions. Experimental observables (even single-molecule techniques) are results of a combination of many individual interaction components such as electrostatic, repulsion, and dispersion. Since forcefields need to be parameterized to individual interaction components, QM calculations are thus the *de facto* standard for forcefield development[32]. In addition to calculating overall interaction strengths, software known as SAPT[33] has been developed to decompose intermolecular interactions into specific terms, such as electrostatics, induction, exchange, and dispersion. This allows for a forcefield to be developed with accurate separation of total energy into terms that more closely relate to the desired physical interaction, a property that, in theory, leads to better transferability of the general force field parameters.

In addition to forcefield parameterization, there have been many efforts to fuse quantum mechanics calculations in an approach referred to as QM/MM. In QM/MM, the forces on a small region of the system (often an active site, or another region of interest) are calculated using QM, while the forces for the rest of the system are calculated using a simple classical forcefield (with corrections for interactions between the QM and MM regions)[34]. In theory, QM-MM attempts to improve upon the accuracy of pure MM based

methods at limited cost and also offer the ability to study chemical reactions where the chemical bonds break and form, something typical classical force fields can not treat. The current best utilization of FEP approaches in protein-ligand binding prediction to an accuracy of around 0.1kcal/mol[35] for sample host-guest systems, significantly more accurate than current classical MD based approaches. However, this improvement comes with significant reductions in performance. Therefore, in order for these approaches to become viable, substantial improvements to QM/MM methodologies need to be made.

Molecular Dynamics

Molecular dynamics-based simulations can be derived from Newton's second law of motion, namely that force equals mass times acceleration. Such simulations provide a statistical ensemble of molecules, from which we can then compute physical and thermodynamic properties such as binding free energy. What differentiates between different molecular dynamics methods is mainly how force is calculated and how the resulting acceleration is integrated. I will begin a discussion of molecular dynamics with the design of various integration schemes, as well as a discussion on temperature and pressure control needed to different ensembles. I will then proceed to discuss the differences between different classical forcefield descriptions of molecular interactions. I will then conclude with an overview of free energy calculation schemes.

The core of integration is how one takes a particle (most often a single atom), a starting velocity, and a force, and determine atomic position after a small increment of time (on the order of 1 or 2 fs). The simplest integration scheme is that of Verlet

integration[36]. In Verlet integration, the positions and velocities at time t+Δt (where Δt is the timestep) can be calculated using the following series of equations:

$$v(t+1/2\ \Delta t)=v(t)+1/2\ a(t)\Delta t$$

$$x(t+\Delta t)=x(t)+v(t+1/2\ \Delta t)\Delta t$$

$$v(t+\Delta t)=v(t+1/2\ \Delta t)+1/2\ a(t+\Delta t)\Delta t$$

While the simple Verlet approach is stable, it does not represent the most efficient integration scheme. The nonbonded forces of a system change over timeframes much slower than the bonded vibrational frequency. Therefore, the stability of molecular dynamics simulations is limited by the stability of bonded integration. Instability in the integration of bonded interactions limits the timestep for the Verlet integrator to around 1.0 fs when hydrogen atoms are involved. Further integrator modifications allow for a longer timestep (Δt), and thus greater simulation efficiency. One common approach is referred to as a multi-time step (MTS) integrator. An MTS approach breaks each large timestep into multiple, smaller timesteps, each of which is integrated similarly to typical Verlet-like integration. However, the slow-evolving nonbonded forces are only evaluated at the larger, outer timestep. Since the nonbonded forces constitute the majority of computational costs, limiting the frequency at which non-bonded forces are updated enables an improvement in performance, with timesteps as long as 2fs possible using this approach (known as r-RESPA[37]). Further increases in timestep are possible by moving some of the mass from heavy atoms to the hydrogen, thereby slowing down bond vibration[38]. This improved stability enables timesteps as long as 3fs, while not changing resulting thermodynamic properties (although kinetics is altered).

The above algorithm, as written, does not include pressure or temperature control. These functions are handled by simulation components referred to as a barostat and a thermostat, respectively. Barostats attempt to maintain target pressure using either a probabilistic approach based on energy (as in the Monte Carlo barostat), or virial (used in barostats such as the Berendsen barostat[39], Nose-Hoover[40], or the Langevin piston[41]). The virial is a tensor defined as the change in energy concerning volume. Given the average of the diagonal of the virial tensor (W) and kinetic energy, an instantaneous pressure can be calculated as $P_{inst} = \frac{1}{3*V} * (2 * KE - W)$. Given a target pressure, the barostat then scales coordinates and box size to bring the system closer to the target pressure. Virial based approaches are often better able to handle systems with densities far from equilibrium compared to Monte Carlo methods. It is standard protocol to run initial equilibration at constant pressure, and then run production simulations at constant volume. A thermostat works by adjusting temperatures by modifying atomic velocities and kinetic energy. Examples of popular thermostats include BUSSI[42] and the Anderson thermostat[43]. Through the combined use of a thermostat and barostat, one can perform simulations under isothermal-isobaric ensemble, or constant pressure and temperature (NPT).

FREE ENERGY FROM MOLECULAR DYNAMICS SIMULATIONS

In addition to the fundamental knowledge gained during molecular dynamics simulations, it is possible to use molecular dynamics approaches to calculate the binding

free energy of a ligand to its target protein. This can be accomplished via one of two classes of approaches - pulling, or alchemical.

Pulling approaches to binding free energy calculation attempt to calculate the binding free energy using an approach in which the ligand is gradually pulled away from the binding site into the surrounding solvent environment using an artificially applied force[44]. The Potential of mean force is then calculated by integrating the applied force magnitude vs. the pulling x coordinate. This work integral is then equal to the binding free energy. The primary difficulty in the utilization of PMF approaches is in finding a useful definition of a pulling coordinate. For some systems, this coordinate is obvious. For example, membrane pores have a clear pathway for the ligand to exit (namely through the hollow pore). Therefore, a multitude of studies has been performed to study channel protein selectivity[45]. However, this pulling dimension is often challenging to define. Most protein-ligand binding pathways are too complex to describe using this pulling approach[46]. Therefore, while a rigorous approach, pulling PMF-based approaches cannot always be utilized.

Much like the alchemists of yore attempted to transform one element into another, alchemical approaches to free energy calculation attempt to transform a simulated system from one state into another. In the case of binding free energy simulations, this consists of transforming the ligand from one that interacts with proteins and water like in the "real-world" to one that does not interact with its environment at all[47]. The energy associated with this change is then the complexation energy (if simulated in a protein-ligand system), or the solvation energy (in the case of a ligand-water environment). The

binding free energy can then be calculated as the complexation energy minus the solvation energy. While one could theoretically calculate either transformation in a single step, converged solutions require that one does this transformation in small perturbations over ~10-20 steps. The energy associated with each transformation step can be calculated using several methods, including the Bennett acceptance ratio (BAR)[48] or thermodynamic integration (TI[49]). Once the energy of each transformation is calculated, one can calculate total binding energy by merely summing up the contributions of the each of the individual transformations since free energy is a state function that is independent of paths. The advantage of alchemical approaches is that unlike pulling approaches, alchemical approaches are universally applicable. The disadvantages of these approaches are a reliable protein-ligand complex structure is needed as input, and computational throughput is relatively low compared to approximated docking, as more individual simulations are needed in order to calculate the final, binding free energy.

Forcefields

A forcefield is a mathematical description of how simulated atoms in a molecular system interact. At the core, most modern forcefields contain similar bonded terms. Bonded interactions are described as a simple pairwise bond term, an angle term, and a torsional dihedral term. Also, the bending of atoms out of the plane and the distortion of pi-bonds are utilized. Where forcefields often differ is in their treatment of the non-bonded forces. Non-bonded forces are often broken up into two major components, a van der Waals term, and an electrostatic term. The van der Waals term attempts to encompass

short-range repulsion and the London dispersion force. The vdW energy is mathematically described as the difference between an attractive and a repulsive term such as $E = 4\varepsilon((\frac{\sigma}{r})^{14} - (\frac{\sigma}{r})^{7})$ with σ representing the separation distance (r) that results in zero interaction energy, and ε representing the energy well depth[50]. There is no theoretical basis for the 14th powers (for repulsion) in the above equation. It is common to see mathematical forms that use alternative powers (a combination of a 12th and 6th power are also often utilized).

More divergence between forcefields arises from electrostatics. Most forcefields represent atomic electrostatics as merely a point charge (monopole). Thus, the electrostatic energy is calculated using simple Coulomb's law, namely $F = k\frac{q_1 q_2}{r^2}$. This is the approach taken by common forcefields such as AMBER[51]and CHARMM[52]. Using this approach, each interaction is trivial to calculate. However, the number of raw calculations needed to calculate each interaction is on the order of the number of particles squared, a scaling law that is insurmountable for large solvated protein-sized systems. Therefore, an approach referred to as Ewald summation is often used to bring the scaling law to NlogN[53].

However, a fixed charge model is too simple to capture electrostatic interactions accurately. Comparisons of QM potentials with fixed charged fittings reveal relative errors of as high as 16% (depending on molecule) while fitting electrostatics to a model containing atomic monopoles, dipoles, and quadrupoles was able to obtain errors of (at most) 0.4%[54], which are utilized by AMOEBA (the forcefield used in work in this thesis).

One example of a type of intermolecular interaction that requires an electrostatic model up to the quadrupole is the interactions of aryl halogens with electronegative groups[55]. For example, aryl halogens are often found in protein-ligand crystal structures interacting with acidic aspartate and glutamate residues[56]. This is contradictory to the expectation that halogens are electronegative atoms with a partial negative charge. The inductive capabilities of the aromatic ring pull charge density away from the plane of the ring, leaving a positively charged patch at the "edge" of the halogen atom, in the plane of the aromatic ring. A simple monopole charge model would assign the halogen a negative charge, and simulations would not exhibit this experimentally identified interaction. However, this electronic distribution can be captured via the dipole and quadrupole terms present in AMOEBA[57]. While this is an extreme example of electronic distributions being better captured via higher-order multipole terms, it illustrates how the dipole and quadrupole terms allow for the capturing of non-uniform electronic distribution, including lone pairs on O and N atoms.

Another phenomenon captured by more advanced forcefields is the polarization effect. Polarization describes the response of electron distribution of molecules to external fields. Given the negative charge of the electron, electronic distribution moves away from other negatively charge sources, and towards positive charge sources. While most molecular dynamics methods do not explicitly simulate electronic distribution, there have been several approaches to capture the polarization effect on electrostatic interactions. One of these is the fluctuating charge model, where atomic charges fluctuate depending on external electrostatic environment[58-59]. This approach is computationally

efficient compared to other polarization methods. However, it suffers from similar accuracy problems to the monopole model of electrostatics- namely; it treats polarization as treating each region of an atom equally, regardless of directionality. Another class of approaches is that of the Drude oscillator, which gives each atom an imaginary negatively charged particle and a positively charged particle, resulting in the formation of a dipole[60-61]. A third polarization method, adopted by AMOEBA, is the induced dipole model. In this approach, each atom produces an induced dipole (separate from the permanent dipole) in response to the external electric field, as well as neighboring induced dipoles. Since induced dipole force is dependent on neighboring induced dipoles, this requires a self-consistent iterative approach, where the induced dipole moments, energy and force are all converged. This iterative approach, combined with higher-order multipoles, means that AMOEBA is as much as 10 times slower than comparable, less complicated forcefields such as AMBER and CHARMM.

GPU Computing

Given the increased cost associated with higher-order multipoles and polarization, it is critical that AMOEBA is implemented in an efficient computation engine. Traditional CPU computation can only efficiently compute AMOEBA forcefields with up to a few CPU cores, limiting the usefulness of this platform to small systems over a short time scale. One solution to accelerate the computational speed is GPU computing. CPU computing relies on a relatively small number of cores (often 4-8) running at a high clock speed (often 2-4 GHz). This high clock speed is sufficient at serial calculations (i.e., one

needs to be computed after another), leading to clock speed is the limiting factor in performance. In contrast, GPUs run many more cores (1000+) running at a slower clock speed (top of the line RTX 2080 GPUs run at a clock speed of slighter higher than 1.5 GHz)[62]. Given a purely serial task, a GPU will be slower than most modern CPUs. However, if a task can be split up into small, independent problems, GPUs can have a massive performance advantage over GPUs. For example, it would take 1,000,000 clock cycles (at minimum, assuming that one addition can be computed in a clock cycle) for a single core of a CPU to sum all of the numbers from 1 to 1,000,000). For a CPU running at 4Ghz, this amounts out to 0.25 ms. In contrast, a GPU based parallel approach to this algorithm with a clock speed of 1.5 GHz, where adjacent pairs of numbers are added, stored, and the process repeated until one cumulative sum is obtained only takes log2(n) cycles. This amounts to only 20 clock cycles or approximately 1.3 μs. Even though the clock speed is noticeably slower, parallelization can result in orders of magnitude improvement in performance. Fortunately, the highest cost of molecular dynamics computations is the calculation of the electrostatic force through a process known as Ewald summation.[53] Parallel implementations of this algorithm have long existed[63], indicating that the most expensive parts of molecular dynamics can be efficiently computed on GPUs. Through the use of parallelization, GPU approaches are capable of obtaining a 30x improvement over CPU only (6-12 cores) computations[64].

One of the major platforms for the running of molecular dynamics simulations on GPUs is OpenMM[65-67]. OpenMM was initially developed by members of the Pande group and consisted of an open-source molecular dynamics engine for a wide range of

forcefields, including AMBER, CHARMM, and OpenMM. OpenMM is coded in OpenCL[68] and CUDA[69] for GPU, as well as for tradition CPU systems. However, especially for calculation using AMOEBA, several prominent features were missing. Inclusion of the newest AMOEBA updates was slow, and barostat and free energy calculation methods were lacking. Therefore, around 2016, a new branch of OpenMM, named Tinker-OpenMM[64], was created to focus on solving these lingering AMOEBA related issues. Tinker-OpenMM, as its name implies, is a member of the TinkerTools family of programs. This C++ codebase has been designed for access through Tinker software, dynamic_omm and bar_OMM for dynamics and Bennett acceptance ratio (BAR) based free energy, respectively. These Fortran codebases then communicate to Tinker-OpenMM, which launches a GPU based simulation that matches the atomic coordinates, parameters, and velocities provided by Tinker input.

Dissertation Overview

The main push of my Ph.D. consisted of the aiding in the development and evaluation of Tinker-OpenMM as a package containing many of the dynamics features present in Tinker CPU. First, I will discuss the implementation of free energy perturbation methodologies in Tinker-OpenMM. I will then demonstrate the application of this technology in the prediction of the binding free energy and structure poses of ligands to the protein kinase Maternal Embryonic Leucine Zipper Kinase (MELK). Finally, I will describe the implementation of virial-based pressure control into the Tinker-OpenMM platform.

TINKER-OPENMM: ABSOLUTE AND RELATIVE ALCHEMICAL FREE ENERGIES USING AMOEBA ON GPUS[64]

Introductory Statements

During the early stages of my Ph.D. (ca late 2015), GPU computation using AMOEBA consisted mostly as a novelty, with limited practical applications. In order to achieve large scale binding free energy calculations in any reasonable amount of time, one needed to run simulations on a supercomputer. Although the University of Texas has one of the best supercomputing centers in the nation in the Texas Advanced Computing Center (TACC), access to supercomputing resources was precious, only to be used for final, production simulations. It was clear that this manner of performing simulations was untenable in the long term if AMOEBA was to evolve into a tool to truly enable drug discovery efforts.

During those times, I was working on using the 4 GPUs we had to run basic Molecular Dynamics simulations using the AMOEBA forcefield. This software only supported basic molecular dynamics simulation, limiting its usefulness. We realized that if this platform could support binding free energy calculations, we could significantly enhance the usability of Tinker, and potentially enable ligand-drug binding studies at scales previously thought impossible. I then set out to port Tinker's alchemical free energy calculation into Tinker-OpenMM. This modification (among others) launched the start of a new era in the practical use of the AMOEBA forcefield. Instead of a reliance on slow CPU based approaches, or precious supercomputing time, one could perform

efficient drug-binding calculations on a GPU architecture that is both high performance and affordable.

Abstract

The capabilities of the polarizable force fields for alchemical free energy calculations have been limited by the high computational cost and complexity of the underlying potential energy functions. In this work, we implement a GPU-based general alchemical free energy simulation platform for polarizable potential AMOEBA. Tinker-OpenMM, the OpenMM implementation of the AMOEBA simulation engine has been modified to enable both absolute and relative alchemical simulations on GPUs, which leads to a ~200-fold improvement in simulation speed over a single CPU core. We show that free energy values calculated using this platform agree with the results of Tinker simulations for the hydration of organic compounds and binding of host-guest systems within the statistical errors. In addition to absolute binding, we designed a relative alchemical approach for computing relative binding affinities of ligands to the same host, where a particular path was applied to avoid numerical instability due to polarization between the different ligands that bind to the same site. This scheme is general and does not require ligands, for which we compute the relative affinity, to have similar scaffolds. We show that relative hydration and binding free energy calculated using this approach match those computed from the absolute free energy approach.

Introduction

Free energy is the driving force for spontaneous molecular processes, and accurate alchemical free energy calculations can benefit a broad range of chemical and biomedical applications[70-74]. The accurate prediction of the binding affinities for ligands to their target proteins has been a significant challenge in the computational drug development process[47]. Today, it is common to utilize empirical docking algorithms in the identification of potential lead compounds. However, in order to screen large ligand libraries in a short amount of time, empirical docking typically relies on crude and inadequate physics models[75], and only account for limited system dynamics (such as loop flexibility) when predicting ligand affinity[76]. These limitations result in a lack of the accuracy necessary for lead optimization[77-78]. The calculation of ligand binding free energies from elaborated molecular simulations has also been limited by a combination of underlying force fields and sampling algorithms[79-80].

One approach for the calculation of binding free energies is the double decoupling scheme. In this approach, one includes a parameter (lambda) that controls the interaction strength of a ligand with its environment, including electrostatics and van der Waals interaction. When gradually transitioning from lambda=1 (full ligand-environment interaction) to lambda=0 (no ligand-environment interaction), a ligand's interaction with its environment is evaluated over the ensemble generated from molecular dynamics, which provides the free energy of alchemical change. Simulations of the system are conducted with the solvated ligand and the protein-ligand complex, and the binding free

energy is calculated as the complexation energy minus the solvation energy, plus standard state and other corrections[81]. In this methodology, restraints[82] are often used to keep the ligand bound to the protein complex throughout the decoupling process. The magnitude of this restraint term is then analytically corrected for.

Another primary class of approaches of binding free energy involves the calculation of the potential of mean force of pulling ligand away from protein target. In these approaches[44], one calculates the average force needed to maintain a system in a given configuration (e.g., the distance and orientation between a ligand and the active site). Free energy is then calculated by calculating the work integral from the starting to ending distances. In order to obtain energy data on all relevant distances, a biasing process such as steered MD[83-84] or umbrella sampling[44] is often used. The advantage of this technique is that it allows for the collection of free energy profiles, including information about the energy barriers to the binding. The main challenge of this approach is the difficulty in defining an appropriate reaction coordinate for the biasing process. Therefore, this technique has been mostly applied to systems such as channel proteins[45, 85] that have an apparent pulling dimension. However, this technique can also be applied to general protein-ligand binding[86-87].

The free energy between the bound and unbound states in either approach can be sampled by using various techniques such as free energy perturbation (FEP)[72], thermodynamic integration (TI)[49], metadynamics[88-90] or Orthogonal Space Random Walk (OSRW)[91-92]. A standard method for calculating the free energy between neighboring states in alchemical perturbation is the Bennett acceptance ratio (BAR)[48]. The free energy

of binding can then be calculated as the difference between the ligand-host interaction free energy and the ligand-water interaction free energy. In thermodynamic integration, one utilizes lambda much like in setting up a simulation for BAR and calculate the numerical integration of $<\partial H/\partial\lambda>\lambda$ from lambda=0 to lambda=1[48]. Compared to BAR, it can be challenging to determine which discrete values of lambda should be used, as convergence can be difficult in regions of high curvature of $<\partial H/\partial\lambda>\lambda$. Due to this, comparison studies[93] have suggested that TI simulations may require more states than BAR to reach converged free energies. However, TI simulations require less post-simulation processing than BAR based approaches.

The second ingredient of free energy simulations is the choice of force field, which is used to model the interaction energy. Popular force fields include CHARMM[52, 94-96] and AMBER[51, 97-99]. More recent advances have resulted in the development of force fields with more complex electrostatics models, particularly the incorporation of polarization and anisotropic atomic charge distributions. General polarizable force fields include polarizable multipole based AMOEBA[100-102], polarizable OPLS[32, 103-104], fluctuating charge[59, 105] and Drude-Oscillator[106-108] based CHARMM force fields. The defining feature of the AMOEBA force field we have been developing is its electrostatic model based on permanent atomic multipoles, as well as many-body polarization through induced atomic dipoles. These added terms, while computationally expensive, allow for more rigorous modeling of ligand-protein electrostatic interaction than is possible using a fixed-charge based force field.

Previous work using AMOEBA force field has shown an accurate recapitulation of experimental free energies in small molecules hydration,[109-112] metal ion hydration[113-115], as well as ligand binding in synthetic hosts[116], and protein systems[102, 117-121]. The inclusion of a complex electrostatic force leads to increasing computational cost so that potentially it can benefit even more from parallel computing of protein-scale systems consisting of tens of thousands of atoms (including solvent water). Earlier implementations of AMOEBA in Tinker have utilized OpenMP[122], which allows for limited parallelism on commercially available CPUs. Massively parallel computation using AMOEBA is possible on supercomputers using the Tinker-HP package[123-124]. Also, AMOEBA has been previously implemented in OpenMM, enabling massively parallel molecular dynamics simulations on Graphics Processing Units (GPUs)[66-67]. In order to enable alchemical free energy calculations in OpenMM on GPU, we have incorporated "lambda" into force and energy calculation via a soft-core approach[125], which is necessary to remove the singularities in vdW interactions that occurs when atoms are in close contacts.[126] Also, we modified the tinker-OpenMM interface to allow for perturbation of the electrostatic force via the scaling of electrostatic parameters. Another feature of OpenMM that is now supported by the Tinker-OpenMM interface is the addition of support for the CustomCentroidBondForce. This addition enables the coupling of two groups of atoms (such as a ligand and its binding site).

Compared to the state of CPU alchemical free energy calculations, GPU alchemical free energy calculations is still in its infancy. It is possible to perform straight molecular dynamics (MD) simulations on GPUs using a few software, including

AMBER[127], NAMD[128], and OpenMM[66]. However, very few GPU platforms have yet supported alchemical simulations. In addition to the work with OpenMM-AMOEBA described here, the YANK package for the use of OpenMM to simulate AMBER force fields is currently in development. Therefore, the AMOEBA force field on GPU implementation described here (Tinker-OpenMM) constitutes the first available platform for free energy perturbation simulations on GPUs using a polarizable force field.

It is not always necessary to compute the absolute alchemical free energy, as the binding or solvation energies relative to a reference ligand are often sufficient. In those cases, it may be advantageous to calculate relative energies in a "perturbative" way, i.e., the ligand in the protein binding site morphing from one to another instead of disappearing completely. The advantage of relative free energy computation is that the host (protein) molecules do not have to go through the apo form, which sometimes may involve large changes in conformational state. Many previous relative binding free energy calculation uses a "dummy atom" single topology approach[129] where a pair of ligands are simulated as a common core of atoms connected to a set of atoms sufficient to describe both desired molecules. This dummy atom approach has been used to calculate several molecular properties, including binding free energies[130-133] Previous work with the AMOEBA force fields on CPUs, have accurately calculated the relative binding free energies of ligands to trypsin using a single topology approach[119-120]. The application of this scheme is, however, not general; it is more suitable for pairs of molecules with significant chemical similarity and sharing a common core. A different approach is to use a dual topology, where two ligands are always present in the binding pocket, and their

interactions with the environment are combined properly: λ*(lig1+pro)+(1-λ)*(lig2+pro). Relative complexation free energy is calculated via a path starting in a state with full ligand 1-environmental interaction and ending at a state of full ligand 2-environmental interaction. Dual topology free energy calculations have been possible in CHARMM since the late 80's[134] and have more recently been implemented in AMBER[127]. However, this dual topology scheme is more challenging to implement in a polarizable force field due to the complexity of the electrostatics (non-additive interactions between the ligands), making it difficult to selectively "scale" the polarization between two ligands. By utilizing a pathway where only one ligand has polarizability during any perturbation step, we were able to avoid this complication.

Currently, the ability to perform GPU based platform alchemical simulations, particularly for polarizable force fields, has been limited. In this work, we created Tinker-OpenMM, an OpenMM implementation of AMOEBA that enables alchemical free energy calculations on GPUs, while also adding the capability to perform dual topology simulations to both the Tinker[135] and OpenMM[66-67] platforms. We then proceed to test the GPU based free energy calculations for hydration free energies of aromatic systems[136], absolute and relative binding free energies of the sampl4 host-guest systems[137].

IMPLEMENTATION DETAILS

TINKER-OPENMM INTERFACE

Tinker-OpenMM is built using an interface to pass tinker coordinates and parameters to OpenMM. Tinker reads in the input key and coordinate files and passes the relevant variables into a C++ script. This script then uses the OpenMM C API to create the relevant OpenMM parameters and forces and initiates GPU Molecular Dynamics simulation. Coordinate saving is then managed by occasionally transferring atomic coordinates and velocities from the GPU to main system memory. Tinker then saves these outputs in Tinker coordinate and velocity files, enabling post-processing by Tinker commands (e.g., BAR). This interface was created by Mark Friedrichs, Lee-Ping Wang, Kailong Mao, and Chao Lu.

ABSOLUTE BINDING FREE ENERGY

In this work, we employ double-decoupling and alchemical perturbation to compute the free energy of binding. First, the electrostatic interactions between the ligand and its environment (water or protein/water) are scaled from 0 to 100% in a series of simulations. With no electrostatic interaction between ligand and surroundings, a series of simulations are run where the (softcore) vdW interactions between ligand and environment are scaled. The path utilized for absolute complexation simulations is shown in **Figure 1.** This process is also repeated in an aqueous environment to account for hydration free energy.

After running these simulations, the Bennett Acceptance Ratio (BAR) method is used to calculate the free energy difference between the two neighboring states. Since energy is a state function, we can calculate the total complexation energy as the sum of many small perturbations in ligand-environmental (protein and water) interaction strengths. The same process is repeated for the free ligand in water to compute the hydration free energy. The binding energy is calculated as the complexation free energy, minus the hydration free energy, with the addition of several corrections, explained below. When conducting alchemical perturbation, it is necessary to denote which atoms belong to the ligand. In the simulation system, the ligand atom indices are identified by using the *ligand* keyword in the key file (e.g. "*ligand* −1 14" denotes that atoms 1 through 14 belong to a ligand). Alteration of the electrostatic interactions between the ligand and its environment is accomplished via the scaling of the electrostatic parameters passed from the Tinker interface to OpenMM. The atomic charge, dipole, quadrupole, and polarizability of all ligand atoms are each multiplied by the current simulation electrostatic lambda value (between 0 and 1), which is denoted by the *ele-lambda* keyword. This results in no electrostatic interaction between the ligand and its environment when *ele-lambda*=0, and full interaction strength when *ele-lambda*=1. This methodology also "turns off" the intra-ligand electrostatic interactions. When calculating hydration free energy, the intra-ligand/solute electrostatic contributions are added back by "growing" the electrostatic parameters for ligand alone (in the gas phase). However, when calculating binding free energy, this contribution is exactly canceled by an equal omission in the ligand-solvent interaction.

When conducting alchemical perturbation simulations, the change in energy and structure that results from each perturbation needs to be relatively small. To avoid the numerical instability of the standard vdW function when the ligand-environment interaction approaches zero, a softcore buffered 14-7 vdW (energy equation shown below) has been used to calculate the energies and forces.[120]

$$U_{ij}^{vdW} = \lambda_{ij}^{5}\varepsilon_{ij}\frac{1.07^{7}}{0.7(1-\lambda_{ij})^{2}+(\rho_{ij}+0.07)^{2}} * \left(\frac{1.12}{0.7*\left(1-\lambda_{ij}\right)^{2}+\rho_{ij}+}-2\right)$$

Here ε_{ij} is the well depth, and ρ_{ij} represents the current interatomic distance divided by r_{min}, the interatomic distance that results in the lowest vdW energy. In order to use this softcore vdW force, we need to assign the appropriate value of the lambda parameter λ_{ij}. In this implementation, each ligand atom is assigned a lambda value equal to the vdW-lambda keyword value in the simulation input key file. Each non-ligand atom is assigned a lambda value of 1. When calculating a pairwise vdW interaction, it is necessary to have a set of combining rules to convert two atomic vdW lambdas into a combined, λ_{ij}. For a pair of atom i and j, λ_{ij} is determined as the lesser of λ_i and λ_j. If the two lambda values are identical (as is the case in an intra-ligand or water-water interaction), λ_{ij} = 1.

In order to ensure that the ligand stays in the binding pocket even when intermolecular interactions are weak, a distance restraint, $k(r - r_0)^2$, is applied between the centers of mass of the ligand and the center of the binding pocket. The bias introduced by the restraint is corrected for at the start and end of our thermodynamic

path. The restraint correction at the end of simulation where no intermolecular interaction between ligand and environment is given by[138]

$$\Delta G_{restraint} = RTln\left(C^0\left(\frac{\pi RT}{k}\right)^{3/2}\right)$$

Here, C^0 represents standard state concentration (1 mol/L). In this work, we use a force constant (k) of 15 $kcal/mol/Å^2$, and this correction amounts to 6.25 $kcal/mol$.

To remove the ligand restraint from the system with full ligand-protein interaction, we repeat the simulation but with the restraint off. The free energy difference between the two simulations is then calculated using BAR. More commonly, one could also gradually turn off the restraint while the interaction strength between ligand and protein increases so that no additional correction is needed.

DUAL-TOPOLOGY RELATIVE FREE ENERGY

Relative binding free energy can potentially be calculated more reliably as it avoids simulation of the no ligand-bound (apo) form of the protein. In this implementation of the calculation of relative binding free energies, we take a thermodynamic path where we first reduce ligand 1's electrostatic parameters (including atomic polarizability) to zero magnitude. We then proceed to reduce the vdW interactions between ligand 1 and the environment, while simultaneously increasing the vdW interactions between ligand 2 and environment. Finally, we increase ligand 2's electrostatic parameters from zero to full. The path we used to calculate relative

complexation energy (ligand binding to the receptor in water) is shown in **Figure 2**. Since the two ligands are never charged at the same perturbation step, ligand 1 and 2 never interact with each other (the vdW interactions are also turned off via the soft-core formula), which requires minimal changes to the electrostatic force in the existing OpenMM code.

To run the simulations in our thermodynamic path, we require independent (ligand 1 and ligand 2) keywords to denote the indices of ligand 1 and ligand 2, respectively. The electrostatic perturbation segments of our path require that we independently control the electrostatic interaction of ligand 1 and ligand 2. This is accomplished by having two electrostatic lambda keywords (ele-lambda1 and ele-lambda2, respectively). The atomic charge, dipole, quadrupole, and polarizability of each ligand is multiplied by the appropriate ele-lambda variable.

When perturbing the vdW force, we need to assign each ligand atom the correct lambda value. The vdW-lambda of all ligand 1 atoms is equal to the value specified by the vdW-lambda keyword, and vdW-lambda of all ligand 2 atoms is equal to 1 minus vdW-lambda. Therefore, changing the vdw-lambda keyword from 1.0 to 0.0 results in removing all ligand 1–environment interactions while setting all ligand 2 atoms to full vdW interaction with the environment.

When conducting relative binding simulations or BAR energy calculations, we need to ensure that the two ligands do not interact via the vdW force. Therefore, we need a way for our vdW force and energy calculations kernels to know which ligand each atom belongs to. This is accomplished by adding an internal variable to the vdW force used to

designate which ligand (if any) an atom belongs to. This variable is equal to 0 for environmental (nonligand) atoms, 1 for ligand 1, and 2 for ligand 2. Each pairwise vdW interaction is checked to ensure that ligand 1–ligand 2 interactions are omitted.

The relative binding free energy is calculated as the relative complexation energy minus the relative hydration energy. Note that if one uses the same force constant for ligand-receptor restraint for all simulations, the restraint correction discussed above is identical for both ligands and drops out in the relative binding free energy.

Methods

SIMULATION SETUP

Before all simulation, the system energy was minimized to avoid close atomic contacts. All simulations were run under OpenMM mixed-precision mode. Ewald cutoff was set to 7.0 Å, with a 12 Å vdW cutoff in both simulations. All simulations converge the root-mean-squared difference in induced atomic dipole moments between iterations to <0.00001 D. Sampl4 and aromatic simulations use a cubic box of 40 Å an Ewald grid of 48 × 48 × 48, while the larger bench7 dataset uses an Ewald grid of 64 × 64 × 64 and a cubic box of 62.23 Å.

MOLECULAR DYNAMICS

Perturbation steps for absolute binding and solvation simulations were conducted with a stepwise reduction of the ele-lambda keyword from 1 to 0, followed by a stepwise reduction of the vdw-lambda keyword while keeping ele-lambda at 0. MD used a RESPA integrator and a BUSSI thermostat.

Relative binding and solvation simulations were conducted starting with the ele-lambda1 and vdw-lambda keywords at 1.0, and the ele-lambda2 keyword at 0.0. In a series of simulations, the ele-lambda1 keyword is then gradually reduced from to 0.0. Simulations follow this with a stepwise reduction of vdw-lambda1 to 0.0, then a stepwise increase of ele-lambda2 from 0 to 1.0.

All CPU simulations were conducted using Tinker program "dynamic" for 1ns with a 2fs time step and snapshots saved every 1 ps. Each GPU perturbation simulation was conducted using "dynamic_omm" for 5 ns, with a 2 fs time-step and snapshots saved every 2 ps (except for relative free energy simulations, where snapshots were saved every 1 ps). All simulations were conducted at 298 K.

BENNETT ACCEPTANCE RATIO

The free energy between steps was computed using Tinker's BAR program. This program iterates between the two equations below until convergence:

$$e^{-\beta\Delta F} = \frac{\langle f(\beta(U_2 - U_1 - C\rangle_1}{\langle f(\beta(U_1 - U_2 + C\rangle_2}$$

$$C = \Delta F$$

$$\text{where } f(x) = \frac{1}{1 + e^x}$$

Typically frames of the initial period of equilibration (~500ps) were ignored.

HYDRATION OF AROMATIC COMPOUNDS

Parameters for the molecules were previously generated.[136] Structures of the 10 compounds are shown in **Figure 3.** Initial simulation systems were generated by

solvating each ligand in water boxes using the Tinker commands solvate and crystal. Initial structures for relative hydration free energy (HFE) simulations were generated by concatenating ligand 2's coordinates to the solvated ligand 1 pose. To calculate the absolute hydration free energy, it is necessary to correct for the contribution of intramolecular electrostatics as we scale the solute electrostatic parameters in "disappearing" or "growing" the solute molecule. To correct this intrasolute electrostatic interaction, each molecule was simulated alone in a nonperiodic system (gas-phase) at ele-lambda values of 0, 0.1, … and 1.0. Stochastic dynamics simulations were run for 1ns using a time step of 0.1 fs, with structures saved every 0.5 ps at 298 K. The intrasolute electrostatic free energy was then calculated using BAR.

SAMPL4 HOST-GUEST BINDING SIMULATIONS

Parameters and starting pose for 12 ligand molecules of the sampl4 dataset were generated as described previously[116]. Structures of the sampl4 ligands utilized in this study are shown in **Figure 4**. The final absolute binding energy was calculated as ΔG of complexation (from no interaction to full interaction) – ΔG of solvation (from no interaction to full interaction) + ΔG of going from no restraint to full restraint at 0 interaction lambda + ΔG of removing the restraint at full interaction energy.

The latest version of Tinker is available at https://github.com/jayponder/tinker. Tinker-OpenMM is available at https://github.com/pren/tinker-openmm. Note that Tinker only works using the modified Tinker-OpenMM, not the main OpenMM release.

Results

FORCE AGREEMENT

Accurate simulation of molecular systems requires an accurate calculation of both force and energy. However, since energy is only utilized by Tinker in the BAR process, and is not used during OpenMM molecular dynamics, we focused our initial analysis of Tinker-OpenMM on the agreement of OpenMM forces with those of Tinker. To ensure that lambda was working in the Tinker-OpenMM implementation, we tested molecule 1 of the sampl4 dataset bound to the host at a range of lambda values and compared the resulting static forces to those of Tinker. The Tinker-OpenMM platform was able to match that of Tinker for all tested lambda values closely, with a root mean squared error of approximately 8.6×10^{-4} kcal/mol/Å, and a maximal atomic force deviation of approximately 4.7×10^{-3} kcal/mol/Å (**Table 1**). These degrees of deviation are negligible when considering that the RMS force is 31 kcal/mol/Å. The force deviation is partially due to the single-precision used in GPU force evaluation.

COMPUTATIONAL EFFICIENCY

To test the speed and scalability of the Tinker-OpenMM platform, we ran 1000 steps of MD on sampl4 system containing molecule 1 (6417 atoms), and the bench7 test case distributed with Tinker (a protein system of 23,558 atoms). For both test systems, the NVidia GTX1070 and GTX 970 were approximately 66-fold and 40-fold faster than an eight-core CPU simulation, respectively (**Table 2**). A single CPU core is approximately 200-fold slower than simulation on a GTX1070 due to the poor core scalability of Tinker utilizing OpenMP. The GPU platform shows better than linear scaling concerning system size, with a 3.7-fold increase in particle number resulting in a 2.4-fold or 2.5-fold decrease in speed on the GTX1070 and GTX970 platforms,

respectively. This better than linear scaling is likely a result of the smaller sampl4 systems being unable to saturate GPU core utilization, as verified by profiling GPU core utilization during simulations. The change of the vdW force to the softcore 14–7 force resulted in no observable difference in speed compared to the kernel used in OpenMM. This was confirmed by running simulations using a version of Tinker-OpenMM that had been modified to utilize a standard, non-softcore 14–7 vdW force without the presence of the lambda parameter in the codebase.

To test the cost of utilization of softcore vdW, tests were run on bench7 with the relative vdW activated by using two water molecules (atoms 9000–9002 and 9003–9005) as "ligands" for the alchemical dual topology process. Both of these waters had their ele-lambda values set at 0.0, with a vdW-lambda of 1.0. This allowed for the activation of dual topology kernels without introducing extra costs. This system was minimized, and a speed test was run as above. This resulted in a speed of 4.68 ns/day on a GTX 970, an approximately 2.5% speed reduction when compared to the absolute simulations. This small cost is only present when doing relative free energy calculations; when no ligand 2 parameter is set, the cheaper absolute vdW kernel is used for force and energy calculation.

Tinker-OpenMM defaults to a utilizing a "mixed" precision mode in all calculations. This mixed-precision mode uses 32-bit floating point calculation for all forces and integrates using 64-bit floating point precision. Due to the poor double floating-point calculation of the consumer GeForce line of graphics cards, the use of double-precision for both integration and force calculation results in an 18.1-fold reduction in performance on a GTX 970.

GPU/CPU ABSOLUTE FREE ENERGY AGREEMENT

As a test of the ability of the Tinker-OpenMM platform to reproduce the results of the Tinker CPU implementation, we performed hydration free energy calculation on a dataset of 10 aromatic compounds, as well as binding free energies on 12 ligands of the sampl4 dataset. Both the solvation (**Fig. 5**) and sampl4 binding datasets (**Fig. 6**) show agreement within the uncertainty of BAR, with R2 values of (0.9924) and (0.9987), respectively. This, along with the static force calculations, provides strong evidence that the GPU and CPU implementations of the AMOEBA force field produce comparable results. The fact that a high degree of agreement is possible even though the GPU simulations were run for 5 times longer (5 ns vs. 1ns at each perturbation step) is an indication that the tested systems converge relatively rapidly.

GPU/CPU RELATIVE FREE ENERGY AGREEMENT

We then proceeded to test the capability of the dual-topology-based relative free energy platform by computing the relative solvation values for the aromatic dataset. For all tested aromatic molecule pairs, the relative hydration free energy values computed from the dual-topology approach and the difference of two absolute HFE simulations showed an agreement within 0.3 Kcal/mol, with an R2 value of 0.999 (**Table 3**). The observed deviation is likely a result of random, nonsystematic statistical error.

Finally, we tested the relative binding prediction of two pairs of sampl4 compounds. The first set of compounds, mol05 and mol06 share similar scaffolds and show agreement in both complexation and solvation to within the uncertainty of BAR (**Table 4**).

The relative binding between molecules 9 and 10 constitutes a more challenging case that cannot be handled using the dummy Atom-based approach due to the lack of a

shared scaffold. Also, this dissimilarity between the ligands may theoretically make convergence more difficult in the intermediate vdW transitions. Nonetheless, the relative binding platform was still able to agree with the absolute platform to within 0.3 Kcal/mol, demonstrating the advantage of the dual-topology platform.

Discussion and Conclusions

This work reports a GPU implementation of alchemical free energy simulation for polarizable force field AMOEBA. The enhanced speed of GPU over CPU will be valuable for applications such as lead optimization. We have shown that the Tinker-OpenMM GPU platform is capable of reproducing the results of Tinker CPU platform, with an approximately 200-fold improvement in computational performance over what is possible on a single CPU core. This usage of GPU computation significantly improved sampling, which should allow for accounting for slow dynamics such as induced fit effects and other local changes in protein structure. Therefore, we expect the better sampling afforded by the GPU-based platform will potentially lead to improved accuracy in ligand binding free energy prediction.

In addition to raw performance, one of the biggest challenges facing the free energy calculation field is the application of techniques to improve sampling of flexible systems to enable convergence with lesser simulation times. One methodology to achieve this increase in sampling efficiency is the calculation of relative binding free energies. Unlike previously utilized dummy atom-based approaches[129-133], the framework presented here is general and does not require a shared scaffold (set of common atoms) between ligands to be utilized effectively. A particular path has been designed to avoid unstable ligand–ligand polarization in the dual-topology approach. We expect that for

flexible protein systems, the dual-topology approach will be more efficient and reduce the time needed for convergence in comparison with absolute free energy approaches.

Concluding remarks

This study constituted an essential advance in the utilization of AMOEBA on GPUs and acted as the public introduction of the Tinker-OpenMM branch. The foundations of AMOEBA were already present in OpenMM; however, binding free energy calculation via alchemical coupling was not present in OpenMM (or indeed, most GPU platforms). Therefore, this work established an effective way to perform polarizable AMOEBA force field-based dynamics simulations and free energy calculations utilizing GPUs.

The new relative binding free energy scheme presented above has not received extensive testing. While it does produce mathematically consistent results for host-guest binding and small molecule solvation, the real advantage of this platform has not been tested. Ideally, we would want to show that this method reaches either more accurate or faster results than the default absolute binding free energy approach. In theory, this method should enable enhanced convergence when performing simple molecular substitutions (such as what occurs during synthetic substitution studies). However, such an advantage has yet to be shown. I would expect this advantage to occur in more dynamic systems, such as in protein-ligand binding. The simple cyclical hosts tested in this study are likely to be too simple to demonstrate this advantage.

Tables

TABLE 1. FORCE COMPARISON BETWEEN THE TINKER CPU AND TINKER-OPENMM GPU PLATFORMS FOR SAMPL4 MOLECULE 1 AT A RANGE OF LAMBDA VALUES.

VDW lambda/ele-lambda	RMSE force (10–4 Kcal/mol/Å)	Max force deviation (10–3 Kcal/mol/Å)
1/1	8.58	4.69
1/0.5	8.59	4.66
1/0.0	8.58	4.71
0.5/0.0	8.58	4.72
0.0/0.0	8.58	4.72

TABLE 2. PERFORMANCE OF TINKER-OPENMM ON NVIDIA GTX1070 AND GTX970 GPUS, WITHOUT THE MODIFICATION FOR RELATIVE BINDING CALCULATIONS, COMPARED TO TINKER CPU RUNNING ON 8 OPENMP THREADS (4X OF SINGLE CPU SPEED).

	GTX1070	GTX970	CPU
mol01(6417 atoms)	20.0	12.2	0.3
bench7(23558 atoms)	8.3	4.8	0.16

Values are in nanoseconds/day.

TABLE 3. COMPARISON BETWEEN THE TINKER-OPENMM ABSOLUTE AND RELATIVE PLATFORM CALCULATION OF THE SOLVATION ENERGY BETWEEN PAIRS OF AROMATIC COMPOUNDS.

	Relative from Dual-Topology	**Difference from Absolute**
Aniline/Benzene	4.2 ± 0.1	4.0 ± 0.1
Adenine/Pyrrole	11.4 ± 0.1	11.3 ± 0.1
Aniline/Adenine	-10.2 ± 0.1	-10.2 ± 0.1
Benzene/3-Methylimidizole	-9.0 ± 0.1	-8.7 ± 0.1
3-Methylpyridine/pyridine	-0.1 ± 0.1	0.0 ± 0.1

TABLE 4. COMPARISON BETWEEN THE TINKER-OPENMM (GPU) ABSOLUTE AND RELATIVE PLATFORM CALCULATIONS OF THE RELATIVE BINDING FREE ENERGY BETWEEN PAIRS OF SAMPL4 COMPOUNDS.

	mol05-mol06		**mol09-mol10**	
	Relative from absolute	**Relative from dual topology**	**Relative from absolute**	**Relative from dual topology**
Complexation energy	44.3 ± 0.1	44.3 ± 0.1	−56.3 ± 0.1	−56.0 ± 0.1
solvation energy	47.3 ± 0.1	47.3 ± 0.1	−68.0 ± 0.1	−68.0 ± 0.1
total ΔΔG	−2.9 ± 0.1	−2.9 ± 0.1	10.4 ± 0.2	10.7 ± 0.1

Figures

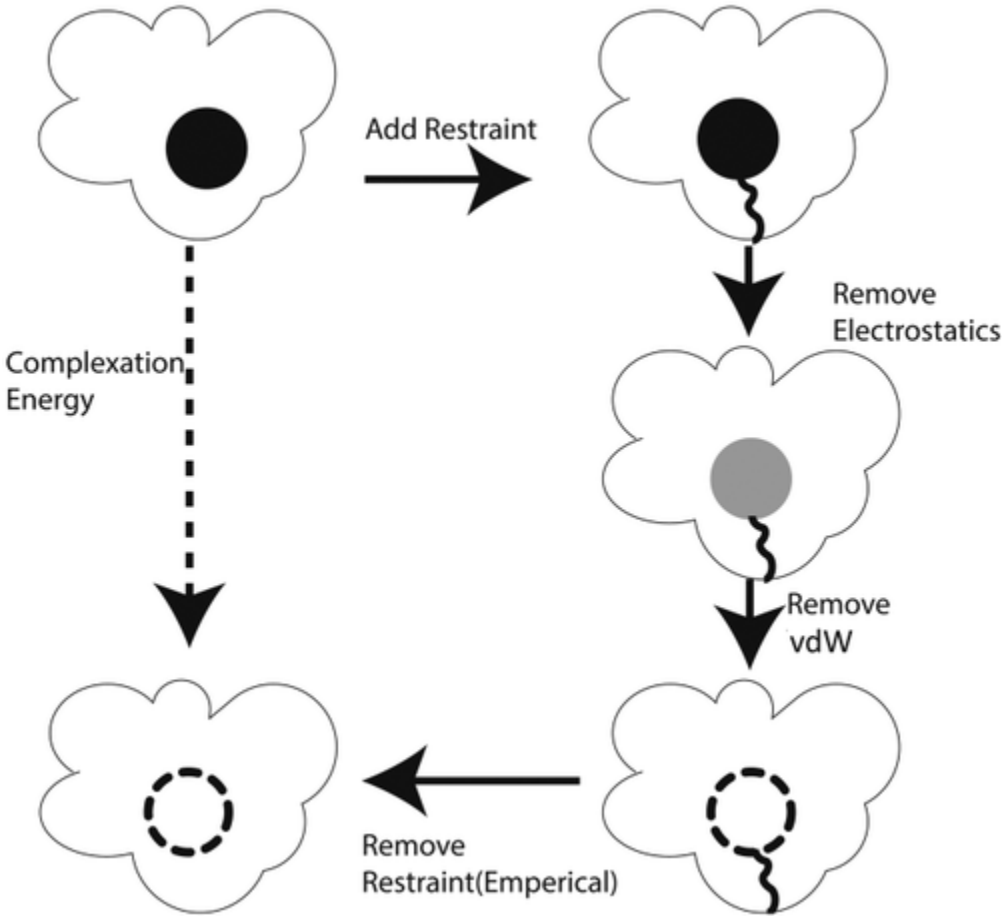


FIGURE 1: THERMODYNAMIC PATH USED TO CALCULATE THE ABSOLUTE COMPLEXATION FREE ENERGY OF A LIGAND USING A DOUBLE-DECOUPLING APPROACH.

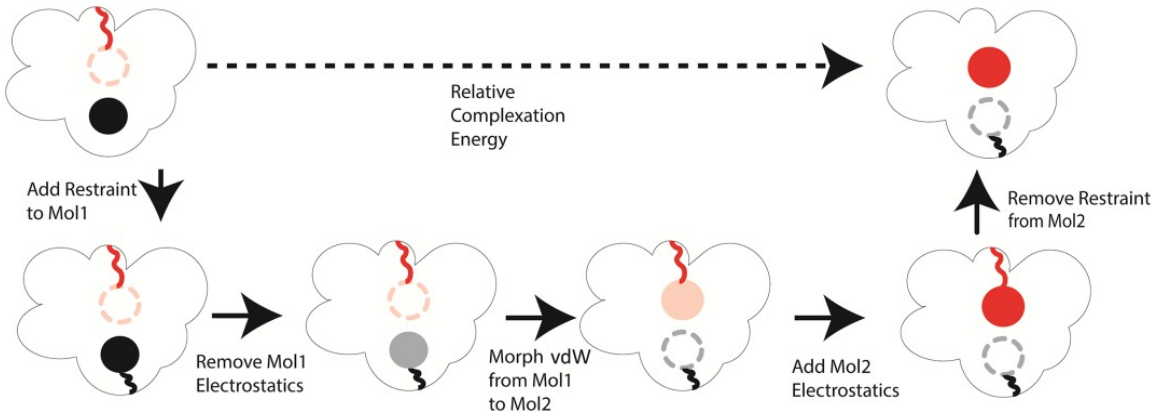


FIGURE 2: PATH USED TO DETERMINE THE RELATIVE COMPLEXATION FREE ENERGY OF TWO LIGANDS USING A DUAL TOPOLOGICAL APPROACH

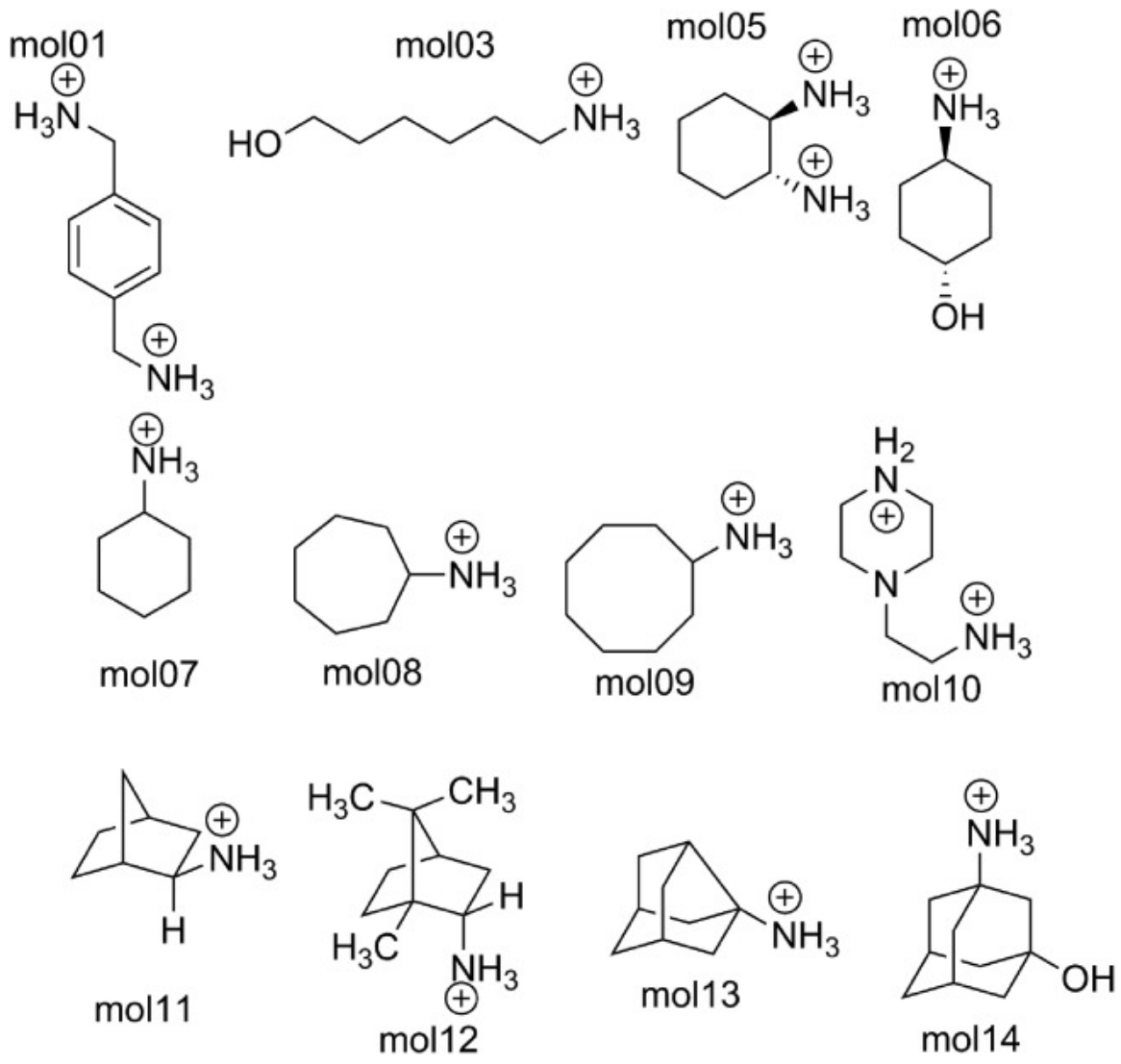


FIGURE 3: STRUCTURES OF THE 12 SAMPL4 MOLECULES UTILIZED IN THIS STUDY.

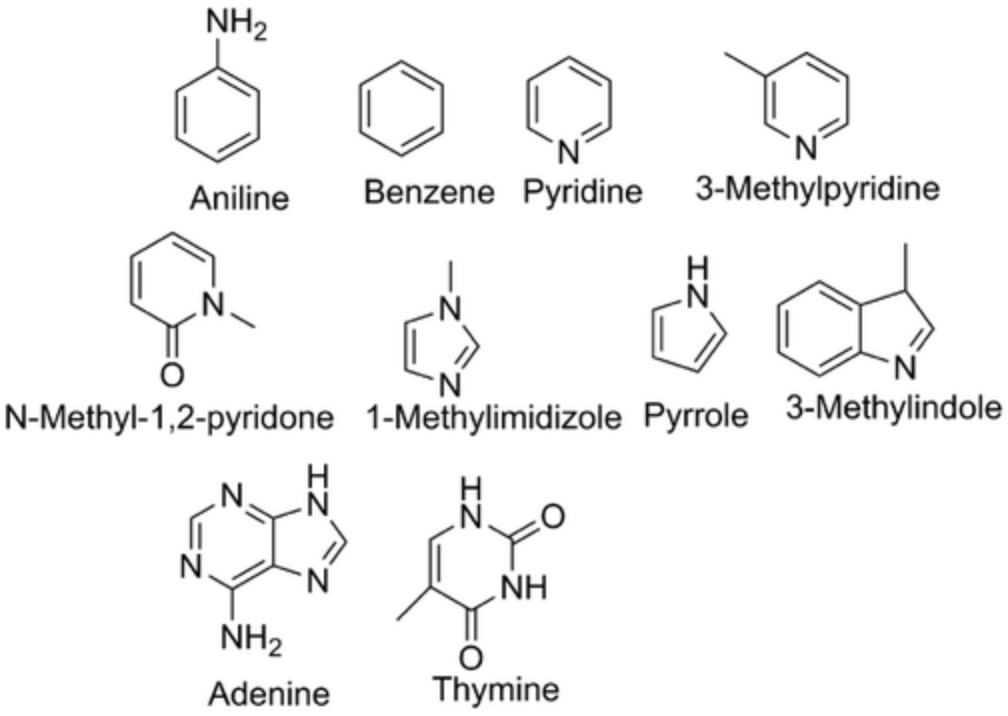


FIGURE 4: STRUCTURES OF THE 10 AROMATIC COMPOUNDS USED IN THIS STUDY.

.

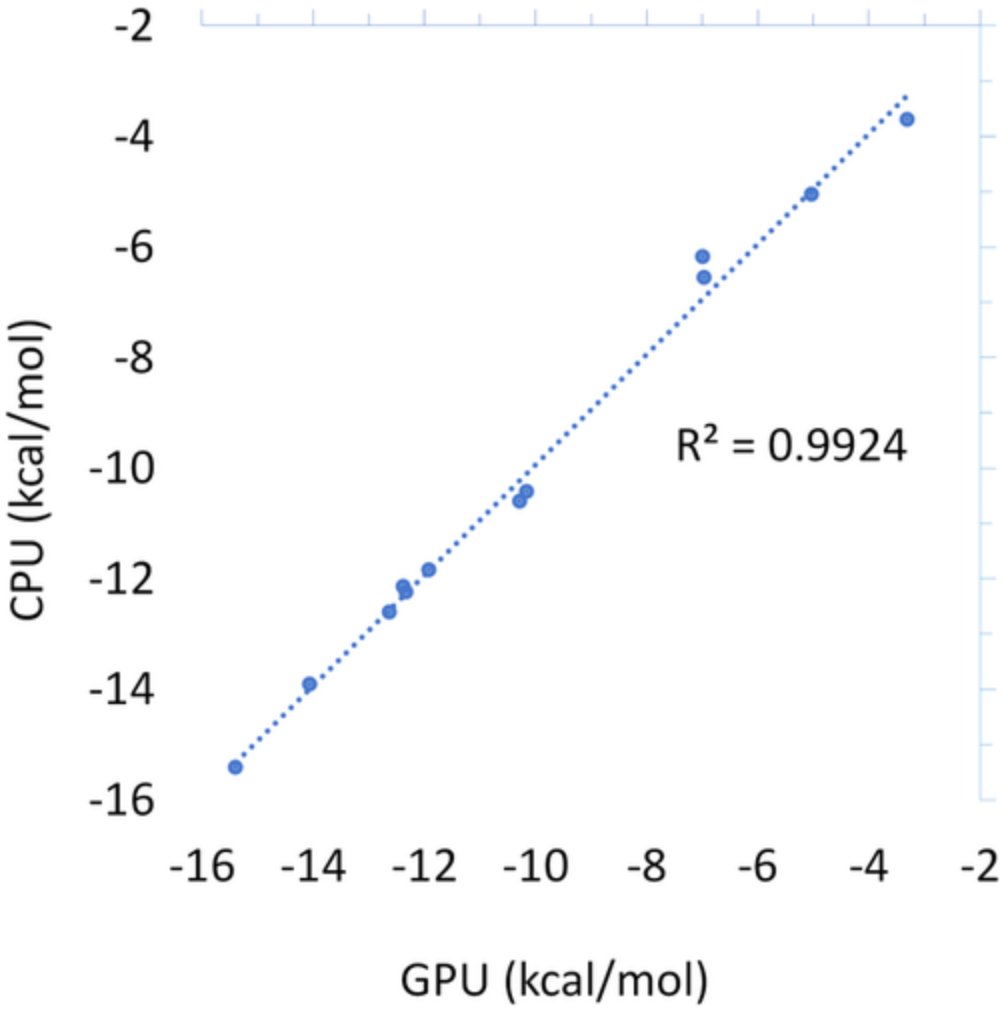


FIGURE 5: COMPARISON BETWEEN THE SAMPL4 BINDING FREE ENERGIES OF 12 SAMPL4 COMPOUNDS COMPUTED BY THE TINKER-OPENMM GPU AND TINKER CPU PLATFORMS. GPU SIMULATIONS WERE RUN FOR 5 NS AT EACH PERTURBATION STEP, WHILE CPU SIMULATIONS WERE RUN FOR 1 NS.

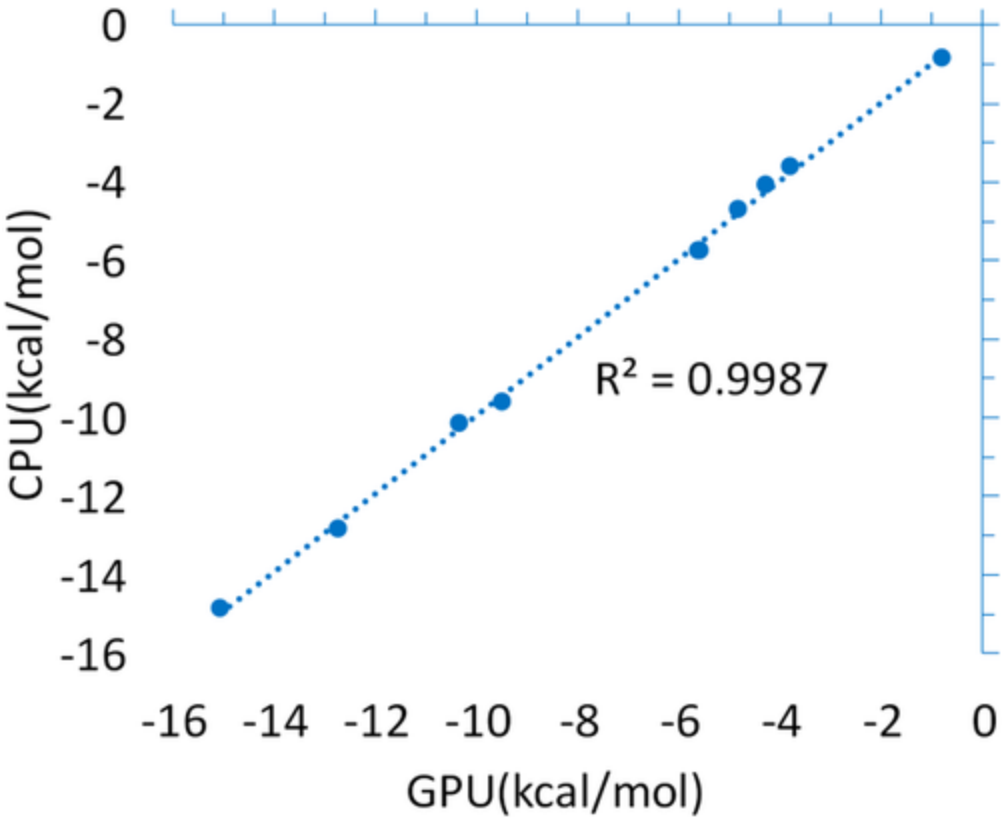


FIGURE 6: **COMPARISON BETWEEN THE CALCULATED SOLVATION FREE ENERGIES FOR THE 10-MOLECULE AROMATIC COMPOUND DATASET ON THE TINKER-OPENMM GPU AND TINKER CPU PLATFORMS.**

COMPUTATIONAL INSIGHTS INTO THE BINDING OF IN17 INHIBITORS TO MELK[139]

Abstract

The protein kinase MELK is essential in cell signaling and has shown to be a promising anti-cancer target. Recent work has resulted in a novel small molecule scaffold targeting MELK, IN17. However, there has been little structural information or physical understanding of MELK-IN17 interactions. Using Tinker-OpenMM on GPUs, we have performed free energy simulations on MELK binding with IN17 and eleven derivatives. This series of studies provide structural insights into how substitution on IN17 leads to differences in complex structure and binding thermodynamics. Also, this study serves as an assessment of the current capabilities of the AMOEBA forcefield, accelerated by GPU computing, to serve as an examination of a molecular dynamics based free energy simulation platform for lead optimization.

Introductory Statements

Many promising anti-cancer targets are protein kinases, enzymes that catalyze the addition of phosphate groups to other proteins[140]. This addition of phosphate causes chemical changes that can have several effects. Firstly, phosphate modification may cause structural changes that result in the activation or deactivation of catalytic or regulatory activity, and thus directly alter cellular behavior. Secondly, this modification target could be a protein kinase itself. Indeed, many protein kinases are interlinked in complex activation and inhibitory networks. This network complexity does several things for a cell. First, it allows for the complex integration of multiple signals. For example, the decision for a cell to divide is dependent on factors such as metabolic state[141], extent, and

integrity of DNA replication[142], and the presence of growth factors[143]. The complexity of these kinase networks allows for the cell to make complex, logic-based decisions, based on any inhibitory and activating signals present.

Also, this complexity makes it hard for the network to be interrupted. Interruptions in proper signaling activity could occur because of the inherent stochasticity of molecular processes, or due to a "bad actor" mutated protein sending wrong signals. When a network is highly dependent on one signal, erroneous behavior is more likely than in an interconnected network in which multiple kinases inform a cellular decision. Therefore, the complexity of kinase signaling cascades acts like a nuclear weapons turnkey; signals from multiple sources need to agree in order to initiate critical cellular events, like division or programmed cell death (apoptosis).

The cancerous state often results from the disruption and rewiring of the cellular kinase network in a way that leads to an activation of processes that lead to cellular division and proliferation, as well as inhibition of cellular apoptotic processes. Given that (at least initially in cancer cell progression), multiple network perturbations are unlikely, it is often the case that cancer is "addicted" to the presence or absence of one cellular signal. For example, proteins in the RAS pathway are mutated in 25-30% of all cancers[144]. This RAS pathway takes extracellular growth factors and transduces as a signal through a kinase signaling network that eventually activates cellular processes needed for growth and proliferation[145]. Mutations in this pathway either cause activation in the absence of appropriate growth signals or ignoring of signals that inhibit signaling. Members of pathways such as the RAS signaling represent promising anti-cancer therapeutic targets. Killing a cancer cell is an easy process. One would need to inhibit a vital cellular component (say, RNA polymerase). The difficulty in anti-cancer drug design is finding a compound that is highly toxic to cancer cells while being minimally

toxic to normal cells. Proteins that cancer is addicted to are enticing targets because of their high expression and importance relative to that in normal cells. This provides the possibility of creating a drug with an excellent therapeutic window that consists of a high enough dosage to harm cancerous cells while still being low enough to have minimal effects of off-target cells.

One such protein that has garnered much interest in recent years is Maternal Embryonic Leucine Zipper Kinase (MELK)[146]. Studies of MELK have revealed that it is highly expressed in many cancers, and its expression is correlated with poor prognosis[147-149]. Due to this these biological studies, a MELK-targeting drug OTSSP167 entered clinical trials. This compound has shown promise as an anti-cancer therapeutic[150-152]. However, further studies reveal that OTSSP167 also targets Aurora B and haspin, two kinases critical for the initiation of mitosis[153]. Inhibition of these kinases would likely lead to some inhibition of cancer proliferation. While OTSSP167 might represent a promising cancer therapeutic, a better understanding of the role of MELK in cancer would require a more specific small-molecule inhibitor.

The IN17 scaffold was discovered by the Dalby lab during an assay of known kinase inhibitors against MELK. During this assay, nintedanib was found to inhibit MELK with a Ki of 5.6 nM, and by merely moving carboxyl tail, the Ki was further improved to 0.39 nM. This is the point at which we started computational efforts to improve upon the structure and knowledge of IN17.

When this project started, no crystal structure of IN17 or a related MELK-ligand complex existed (indeed, a structure of IN17 bound to MELK is still not available). A structure of nintedanib bound to MELK was released several months into this project. While this acted as a valuable opportunity to confirm that the structure of our simulations was correct, it still left many questions about the structure of the IN17-MELK complex

unanswered. The isomeric state of the carboxyl tail, the importance of the interaction of piperazine with E15 residue and the possibility of the interference of HEPES (buffer). These issues stymied a reliable interpretation of computational results. Despite this limitation, this study encompasses a powerful demonstration of the capabilities of the Tinker-OpenMM GPU computing engine, raised various relevant questions that could have been neglected without the modeling effort. Through a retrospective analysis of the results of MELK derivative binding, we show that Tinker-OpenMM is capable of providing 1 kcal/mol accuracy in ligand binding free-energy prediction while also revealing valuable structural insights.

Introduction

The protein kinase maternal embryonic leucine zipper kinase (MELK) has received interest as a potential therapeutic target for cancer. MELK is reported to activate the cancer-promoting transcription factors FOXM1[154] and c-JUN[155] directly and upregulate the expression of the anti-apoptotic protein MCl1 through eIF4B signaling pathway[156]. MELK expression is upregulated in many types of cancer cell cultures and tumor samples[147-149]. Overexpression of MELK is a correlate of poor prognosis in many cancer types, including triple-negative breast cancer[157-158], prostate cancer[159], lung adenocarcinoma[148], and acute myeloid leukemia[160].

Given its potential as a therapeutic target, several inhibitors of MELK have been developed, most prominently OTSSP167[150-152]. However, OTSSP167 exhibits significant off-target binding and has been found to inhibit the mitotic kinases BUB1 and Haspin, as well as Aurora B kinase[153]. Given the importance of these kinases in initiating mitosis[161-

[163], likely, at least some of the therapeutic effects of OTSSP167 are not a result of MELK inhibition. This has made probing the actual role of MELK in cancer progression difficult.

In an attempt to create a more specific chemical inhibitor of MELK, the IN17 scaffold was developed[146]. This scaffold is present in the clinically approved drug nintedanib[164] and was slightly modified by moving the carboxymethyl ester from C29 to C28 to form IN17 (**Figure 1**). IN17 has been shown to bind MELK with a sub-nanomolar K_i, as well as to suppress cellular proliferation in cultured Triple Negative Breast Cancer cell lines[146]. However, binding structural information is lacking for this compound and its derivatives, limiting the potential development of further improved compounds. In this paper, we use molecular dynamics and free energy methods to analyze the binding mechanism of IN17, and related derivatives, to MELK.

There has been a recent revival of interest in the toolkit of protein-ligand binding free energy calculations[165]. The long simulation runs necessary to calculate binding free energy has long been possible in fixed point charge based forcefields such as AMBER[51, 97-99]and CHARMM[52, 94-96]. However, these forcefields have not been able to reliably modeling highly charged compounds (like IN17), or accurately predicting binding free energy consistently[2]. This suggests that much work on improvements to forcefield and sampling schemes is needed for physics-based simulation to reach its full potential.

One approach to improve upon the accuracy of fixed charge models is to utilize polarizable force fields such as AMOEBA[100-101, 166]. The AMOEBA forcefield is characterized by the inclusion of electrostatic polarization via induced dipoles, as well as

the addition of atomic dipole, and quadrupole electrostatic terms. Previous studies have utilized the AMOEBA forcefield to calculate the hydration free energy of small molecules[110-112] and metal ions[113-115], in addition to ligand binding free energy to synthetic hosts[116] and proteins[117-121, 166]. However, until recently, the computational speed of AMOEBA has been a limiting factor for ligand throughput and sampling. The recently developed Tinker-OpenMM platform enables a 200 fold enhancement over what is possible in a single CPU processor through the use of GPU computation[64]. In this study, we have utilized the Tinker-OpenMM platform to perform protein-ligand binding studies at a scale that was infeasible using previous CPU approaches. Given the large size and highly charged nature of the IN17 ligands, we expect the polarization, dipoles, and quadrupoles present in AMOEBA are necessary for accurate modeling.

Methods:

LIGAND PARAMETERIZATION

Initial parameters for IN17 and Nintedinib were generated using POLTYPE[101]. Torsion parameters for all rotatable bonds were derived by fitting to Gaussian 09[167] QM energy at MP2/6-31G* in the gas phase. These rotatable bonds were entered into the valence.py file provided in POLTYPE, enabling the parameterization of IN17 derivatives without recalculating these torsional parameters. The IN17 derivatives were then parameterized using POLTYPE with this new torsional dictionary. In order to speed up the structural optimization of IN17 and derivatives, POLTYPE was modified to run initial structural optimization at wB97XD/6-31G*.

SIMULATION PARAMETERS

Unless otherwise noted, all simulations were run using a 3.0 fs time step with the heavy-hydrogen option in order to increase stability at this longer time step. This keyword moves some of the mass from the heavy atom to the hydrogen[38]**.** MD Frames written out every 2ps. All simulations used the r-RESPA integrator and the BUSSI thermostat (298K). All constant pressure simulations were conducted using the Monte Carlo barostat. All binding simulations utilize a harmonic restraint between the G2 moiety to the centroid of a group consisting of I16 and Y87, which is turned on gradually as the interactions between ligand and surrounding is decoupled (more details in binding free energy simulation discussion). The restraint uses a reference distance of 4.7 Å and maximal restraint constant of 15 kcal/mol/Angstrom (see SI).

COMPLEX STRUCTURE GENERATION:

The initial guess for the MELK structure with bound nintedanib was generated using 4BXY, docking nintedanib into the binding pocket using GOLD[168] at default settings. The resulting complex was minimized to 10.0 kcal/mol/Å with polarization off to resolve clashes, and again at 1.0 kcal/mol/Å with polarization back on. We then ran simulations for 0.3 ns at each temperature from 25-298K under 1 atm pressure, with temperature increasing at 25K intervals, followed by 10ns at 298K with constant box size to equilibrate the system. After the release of the PDB ID 5MAF crystal structure of the complex, we prepared this structure for simulation in a similar manner. 5MAF has a gap in crystal density between Residues 146-177. Therefore the crystal structure PDB ID 4IXP[169] was used to help resolve this extended loop gap between residues 156 and 171 of

5MAF using MODDELER[170]. The MELK-IN17 complex system was solvated in an 84.8Å x 65.2 Å x 65.2Å box of water using the Tinker "xyzedit" command. 2 Mg^+ ions, 41 Cl^- ions, and 22 K^+ ions were added to the water box at random locations to match experimental conditions. This loop was then heated as described above, with all atoms frozen except the modeled loop. This structure was then heated again as above without these added restraints to produce an equilibrated structure. The solvation phase of the free/unbound ligand was generated by soaking the ligand in a 59.8 Å x 46.6Å x 46.6Å equilibrated box of water using xyzedit, adding 1 Mg^+ ion, 10 K^+ ions, and 17 Cl ions to this box.

BINDING FREE ENERGY SIMULATIONS

To generate initial structures of MELK-IN17 derivatives, the structure of IN17 generated above was manually derivatized using Avogadro[171] by editing the IN17 ligand. Avogadro maintains rotational and translational frames, enabling the superposition of the generated derivatives onto apo-MELK. The structures of derivatives were put back into both the protein-solvent system with the water box generated above to produce initial structures of the complex and solvation systems for all the derivatives. The simulation systems were minimized to resolve steric clashes. These complexes were then simulated for 3ns at a constant volume and temperature at 298K in a series of simulations with electrostatic lambda, which scales the electrostatic parameters of the ligand, gradually from 1.0 to 0.0, followed by a series of simulations with vdW-lambda, which scales the vdW interactions between ligand and surrounding using a softcore approach, from 1.0 to

0.0. The exact lambda values for binding phase simulations and solvation phase simulations are available in SI of the publication. The change in free energy, entropy, and enthalpy for neighboring steps was calculated post-MD using Tinker "bar" program, using frames 150 to 1500. The correction due to the distance restraint and standard concentration was calculated using Tinker "freefix" program, which equals 1.38 kcal/mol. The binding free energy was then calculated as ΔG of complexation - ΔG of solvation + the correction described above.

IN17 SOLVENT PHASE CRYSTAL STRUCTURE

MELK-In-17 was dissolved in 5% methanol in dichloromethane in a vial. The vial was wrapped with aluminum foil; small holes were made to the foil. The solution was allowed to sit for 3 days to give crystals suitable for X-ray crystallography. Crystals grew as long, colorless needles by slow evaporation of methanol in dichloromethane. The data crystal was cut from a larger crystal and had approximate dimensions; 0.27 x 0.05 x 0.05 mm. The data were collected on an Agilent Technologies SuperNova Dual Source diffractometer using an μ-focus Cu Kα radiation source (λ = 1.5418Å) with collimating mirror monochromators. A total of 583 frames of data were collected using with a scan range of 1° and a counting time of 23 seconds per frame with a detector offset of +/- 42.4° and 70 seconds per frame with a detector offset of +/- 110.4°. The data were collected at 100 K using an Oxford 700 Cryostream low-temperature device.

Results/Discussion

ARYL-CARBONYL ISOMERISM

During initial structural studies of IN17, we realized the possibility that the C28-C30 bond (**Figure 1**) of nintedanib (as well as IN17) has a partial double bond character. Thus, there is a possibility of two distinct conformational isomers (cis vs. trans) due to the rotation around this bound, likely leading to different net binding energies. Indeed, simulations predict an approximately 1kcal/mol difference in binding free energy between the two carboxyl isomers. In order to determine if these two isomers can readily interconvert, we calculated the quantum mechanical rotation barrier of the C28-C30 bond of IN17. QM calculations predict an 8 kcal/mol barrier of rotation in solvent (using polarizable continuum method or PCM[172]), and a 14 kcal/mol in the gas phase (**Figure 2**). This barrier would be largely inaccessible at room temperatures, indicating that once synthesized, this group is unlikely to swap between the two carboxyl isomers. In order to determine the most likely isomeric state of the carboxyl tail, a solvent phase crystal structure of IN17 was determined (see SI section of publication). This crystal structure displays a well-resolved carboxyl tail, indicative of only one isomer being formed in solution. Similar isomerism may exist in other drug compounds, limiting potency. Further research is required in order to test this hypothesis.

MELK-NINTEDANIB COMPLEX STRUCTURAL PREDICTION

To date, no crystal structure of the MELK-IN17 complex exists. On the other hand, nintedanib is a well-studied MELK inhibitor [173-174] that differs from IN17 only in

the location of the carboxyl tail on the indole ring (in nintedanib the carboxyl tail is attached to C29 in **Figure 1**). We first modeled the MELK-nintedanib complex structures by using virtual docking and Tinker-OpenMM molecular dynamics simulations. Using GOLD, nintedanib was docked into the only ligand-bound crystal structure of MELK available at the time (PDB ID 4BKY[175]), which was then used as a starting point for 10 ns of MD simulations, as described in the methods section. A MELK-nintedanib structure (PDB ID 5MAF[176]) was released after our initial simulations. In this crystal structure, the nintedanib carboxyl ester exists in a configurational state consistent with one of the isomers discussed above. The structure of MELK in 5MAF is in good agreement with the end state from Tinker-OpenMM simulation, with a Cα RMSD of 1.5 Å (**Figure 3a**). Overall, the ligand and binding site residues from simulations adopted poses similar to those in crystal structure 5MAF (**Figure 3b**). This is an indication that the AMOEBA forcefield can capture realistic protein-ligand complex structures for this class of compounds. However, one significant discrepancy was observed between the modeled nintedanib-MELK complex and the newly released crystal structure 5MAF. N1 of the piperazine moiety of nintedanib, rather than being free in solution as predicted by docking and MD simulations based on 4BKY, was bound to residue Glu14 in the 5MAF. This interaction was missed in the initial modeling, as this N-terminal region was not resolved in the 4BKY crystal structure. While 5MAF shows that the piperazine of nintedanib is interacting with Glu14 residue in the crystal, the relevance of this interaction in solution, where the buffer and solvent conditions are different, has not been established. Further discussion of this interaction is presented below.

ABSOLUTE BINDING FREE ENERGY OF MELK WITH IN17

Predicting the absolute binding free energy computationally is more challenging than predicting the relative affinities, where stronger error cancellation often occurs. First, we wanted to determine this pipeline's capabilities in predicting the absolute binding affinity of IN17. Initially, before the release of the crystal structure of the MELK-nintedanib complex, we utilized a MELK-IN17 complex structure, predicted using docking to MELK as a starting point for molecular dynamics and free energy simulation. Simulations based on PDB 4BKY lacked the first 20 residues, including Glu14. This series of simulations resulted in binding free energy of -12.4±0.1 kcal/mol, in reasonable agreement with experiment (-13.3 kcal/mol).

When a crystal structure of the MELK-nintedanib complex (PDB ID 5MAF) was released, this structure was used to generate a MELK-IN17 complex by removing the carboxyl tail and manually adding the carboxyl methyl ester to the C28 position. The predicted MELK-IN17 complex was then used as a starting point for free energy simulation. The main difference is an additional interaction between the positively charged piperazine group of the ligand and the negatively charged Glu14, observed in the crystal structure. One uncertainty is the protonation state of the piperazine moiety. The nitrogen near the terminal of the ligand (N1 in **Figure 1**) is more likely to be protonated due to the inductive effects of the carbonyl group (C7=O1). Simulations of the MELK-IN17 complex in this charge state results in strong Glu14-IN17 interaction and binding free energy of -18.3±**0.2** kcal/mol, 5 kcal/mol more negative than the experimental result of -13.3±0.1 kcal/mol. On the other hand, if the piperazine is deprotonated at the N2

position, this interaction between piperazine and Glu14 virtually disappears, giving binding free energy of -13.7±0.2 kcal/mol, in good agreement with experiment (-13.3±**0.1** kcal/mol). There is a possibility that the Glu14-piperazine salt-bridge interaction may not be essential or present in solution, as opposed to in the crystal lattice.

Also, the experimental measurement was performed at very high buffer concentration (50mM vs. 10nM for protein concentration), which can affect the interaction of this pair due to buffer agent (HEPES) being able to bind in the protein pocket[177]. Note that the HEPES or 4-(2-hydroxyethyl)-1-piperazineethanesulfonic acid, also contains the same piperazine moiety. It is therefore expected to compete with the piperazine group in IN17 when binding to Glu14, especially given the buffer agent concentration is several orders of magnitudes higher than that of the ligand. We computed the binding free energy of HEPES to Glu14 to be -8.4±0.2 kcal/mol. The calculated absolute binding free would be in agreement with experimental measurement if we take into account the protonation state and/or buffer competition. Nonetheless, for the relative affinities among IN17 and its derivatives, the contribution of this piperazine group cancels and becomes irrelevant. Also, this overprediction of affinity could be due to modeling too much average charge on the piperazine nitrogen, or due to misprediction of the protonation state of the piperazine group.

IN17 BINDING MODE

Most of the close interactions present in the IN17 binding mode (≤0.3 nm in **Figure 7**) are also observed across all derivatives in our simulations. The protein-ligand

contacts are mostly between hydrophobic groups, with the relative positioning of Ile16, Gly17, Ala37, and Leu138 serving to provide a tight groove for IN17 binding (**Figure 4**). Other than the Glu14 interaction described above as a potential point of electrostatic contact, few strong electrostatic contacts are present. Cys88 forms hydrogen bonds with the ligand atoms O2 and N4, constraining the relative orientation of the G2 and G3 of the ligand. HN5 of the indole group in IN17 forms a hydrogen bond with the backbone carbonyl of Asp86, but this interaction is unlikely to add specificity. The ester carbonyl tail (COOC) mainly interacts with Lys39, with some hydrophobic interaction with Val24.

RELATIVE BINDING FREE ENERGY OF IN17 DERIVATIVES

After gaining an understanding of IN17's binding mode, we wanted to determine the effects of compound derivatizations (chemical modifications) on ligand binding. In order to attempt to add electrostatic contacts and potentially improve affinity and selectivity, electronegative groups were added to the central polar benzene moiety (R1 and R2 in **Table 1**). Since the meta and para positions of the central benzene ring (G2 in **Figure 1**) are pointed towards the protein and did not appear to have severe steric constraints, the meta and para positions on this ring were chosen for derivitization. Also, the importance of the carboxyl tail in IN17 (R3 in **Table 1**) was not well understood, so we performed studies where the carboxyl tail was removed or lengthened. Since it was uncertain if the piperazine ring was binding E14, the arbitrary decision to proceed with calculations as if this interaction was occurring was made. Since all substitutions were at

positions of IN17 far away from the piperazine group, binding energy relative to IN17 should be unaffected by this decision.

Overall, experimental affinities relative to IN17 were predicted with an optimal RMSD of 0.8 kcal/mol, a raw RMSD of 1.1 kcal/mol (**Table 2**), and an R^2 value of 0.75 (**Figure 5)**. The Kendall's tau (a measure of the relative rank order of compounds) was 0.50. This level of accuracy is sufficient to determine which compounds are unlikely to bind effectively to a target protein, such as in compounds 22 and 23. This ability to predict non-binding compounds would potentially allow for prediction of compound selectivity across a range of related proteins.

THE N-TERMINAL LOOP STRUCTURE IS ALTERED BY SUBSTITUTION ON THE BENZENE (G2) OFFSHOOT

Substitution at the central benzene ring (G2 in **Figure 1**) can result in alterations in a neighboring beta-sheet structure near the binding pocket (**Figure 6**). In IN17 simulations, this beta-sheet is shortened by a bulge that results in hydrophobic packing against the exposed edge of the G2 benzene ring. Interestingly, in the nintedanib structure (PDB ID 5MAF), this loop bulging is not observed, suggesting that the crystal structure of nintedanib provides an inaccurate representation of certain aspects of loop dynamics for IN17. Another explanation is that the lack of loop bulging in 5MAF could be a result of cross-crystal contacts (this loop is surface exposed). When electronegative groups are added directly to atom C21, as in derivatives 18a, 18e, 18g, 18i and 18p, this region forms a beta-sheet, with interactions occurring between the electronegative atom and HN of Thr18 (**Figure 6**). When a carboxyl ester is added to the C21 position (as in

ligand 18d), the beta-sheet structure distorts into a loop to form interactions with the carbonyl oxygen of the carboxyl methyl ester substitution group. The observation that a beta-sheet is not formed in the 18d complex is likely due to rigid structural requirements for the formation of this beta sheet-ligand interaction. The protein beta-sheet structure is rigidly defined, as is the relative positioning of C21 and the neighboring indole group. This rigidity results in not enough backbone or ligand flexibility to form this backbone-ligand interaction unless the para carbon (C21) is directly connected to an electronegative atom. This rigid structural element combined with the knowledge of this alternative beta-sheet form should result in improved ability to predict the structural effects of substitution on this ring.

EFFECTS OF SUBSTITUTION ON THE BINDING MODE

Compared to IN17, the substitutions mainly resulted in only minor changes in contact distance, with most interactions being maintained across all derivatives (**Figure 7**). The exceptions to this are mostly residues Gly17 and Gly91, both of which maintain close contacts in IN17, but not in many of the tested derivatives. Interestingly, the neighboring Ile16 is a strictly maintained interaction, indicating that the alteration of the structure of the loop containing Gly17 is minor, and indicating that interactions between the ligand and Ile16 are likely essential for IN17 and derivative binding. Gly91 is proximal to the derivatized C28, so alterations in structure in this region is expected. This is consistent with the substitutions at this group leading to alterations to the first shell of contacts around this ring, but only minor alterations occurring at other interaction sites.

Any induced fit effects are likely to occur at timescales longer than effectively simulated using the AMOEBA forcefield.

USE OF RESTRAINED EQUILIBRATION TO IMPROVE PREDICTION

Compound 18g represents a case were Tinker-OpenMM poorly predicted the binding free energy, with a relative prediction of 2.1±**0.2** kcal/mol, significantly weaker than the experimental -0.5±**0.1** kcal/mol. We hypothesized that this error was because the equilibration procedure was unable to capture the induced-fit effects involved in the fitting a methyl ether at the meta position, and thus resulted in an unstable pose. If this is the case, further restraining the ligand within the protein pocket and then running a more extended equilibration simulation may result in a more stable starting configuration for free energy calculation.

In order to test the hypothesis, the 18g starting point was equilibrated for 4ns with a 3.0 kcal/mol/angstrom restraint between the terminal methyl carbon of Ala37 and O2 of 18g, as well as between the nearest terminal methyl carbon of Val24 and C15. Both of the restraint distances were set to 3.5 Angstroms. Since the indole moiety of this ligand is tightly bound, and both of these ligand atoms are nearby the R1 substitution point, this region of the ligand is closest to the system instability that resulted from R1 substitution. This end-state was then used as a starting point for free energy simulation, with a gradual reduction of these restraints in the first 6 simulation steps, as well as a 2-step reduction of restraints at full interaction strength (ele and vdw-lambda=1). Thus, the overall simulation end-states are identical to before, while the intermediate states now utilize

additional contact restraints. This series of simulations resulted in a reduction of error in the relative binding free energy from 2.6 kcal/mol to 1.6 kcal/mol, suggesting that additional equilibration with contact restraints can improve prediction for derivatives with strong perturbations. This study illustrates the importance of starting structures for free energy simulations due to the limitation of sampling capability. Further research is necessary on the general applications of contact restrain in free energy perturbation.

ENTROPY-ENTHALPY COMPENSATION

Post-processing analysis of the free energy calculations enables an estimation of the enthalpic and entropic components of binding and solvation energies. Both binding and solvation entropies and enthalpies displayed a wide range of absolute values across the derivative series (Supplementary Table S2 in publication), indicating that even these small changes to ligand structure can result in massive changes to both entropy and enthalpy components, even if the final, binding free energy has limited changes. This entropy-enthalpy compensation analysis also provides insight into why compounds with extended carboxy tails like compound 22 display relatively weak binding. The electronegative tail results in strong enthalpic interactions with MELK, indeed, the -150.8k±43.6cal/mol binding enthalpy is 56 kcal/mol more negative than IN17 and shows unfavorable enthalpic interactions with water (only -34.8±31.3 kcal/mol). However, the entropy losses associated with binding are significantly more negative than that of IN17, resulting in a ΔG that is less favorable than that of IN17.

Also, the relative entropy-enthalpy differences between IN17 (with the carboxyl tail) and compound 16 (without the tail) reveals significant thermodynamics contributions of this carboxyl tail to the binding. Unexpectedly, the entropy change of binding ($T\Delta S$) is much more significant for compound 16 (-76.4 kcal/mol vs. -23.0 kcal/mol for IN17). This cannot be easily explained by ligand entropy alone; one would expect constraining a large group would result in a more significant entropy decrease. As expected, the presence of a carboxyl-ester tail in IN17 results in a significant change in solvation entropy relative to compound 16 (-115.1 vs. -77.4 kcal/mol $T\Delta S$), due to the presence of hydrophobic groups. Comparing IN17 to ligand 16, the 40 kcal/mol increase in $T\Delta S$ almost precisely cancels the 37 kcal/mol increase in solvation enthalpy, and thus the overall binding free energy remains similar. The importance of interfacial waters is emphasized in the apo-MELK crystal structure(5TWU), which contains many structural waters in this pocket, indicating that this pocket is solvent-exposed. Another interesting question is why IN17 displays a much lesser binding enthalpy than compound 16 (-94.87±32.9 kcal/mol vs. -149.8±32.8 kcal/mol). IN17 likely disturbs the apo residue contact network, resulting in a loss of protein-protein contacts that is greater than the gain in protein-ligand contacts. For example, as explained above, the N-terminal beta-sheet is disrupted, causing a loss of protein hydrogen bonds without regaining strong electrostatic interactions. Thus, due to the entropic effects of binding and solvation, as well as disruption of the native protein contact network, the carboxyl group of IN17 causes little improvement in binding affinity vs. compound 16.

This series of simulations provide insight into the importance of entropy-enthalpy compensation. An increase in binding enthalpy is often, although not always, countered by a corresponding decrease in binding entropy. These simulation results illustrate that the exact magnitude of this change is incredibly challenging to predict based on chemistry/structure alone. While one can estimate potential enthalpic interactions, without dynamics information, predicting significant entropic effects is difficult, as are the effects of ligand binding on protein interaction networks. Computational predictions such as those performed in this study allow for an analysis of these effects in a way that cannot be easily assessed by experiment.

Conclusions

The state of computational free energy prediction technologies has reached a point where it can serve as a valuable addition to commonly used experimental and crystallographic approaches for the study of ligand binding structure and thermodynamics. To crystallize the number of derivatives utilized in this study would be highly costly and time prohibitive. However, molecular modeling techniques provide the ability to understand the structural effects of ligand derivatization of the ligand-protein complex in a matter of days. Even in cases where the crystal structure is present, these structures ignore the dynamics of the system, which is quickly captured by molecular dynamics. This study has found many valuable insights into the binding mode of IN17 to MELK, including the importance of carboxyl tail isomerism, and the N-terminal loop/beta-sheet interconversion. The application of free energy simulation technology should enable

more effective and efficient lead optimization, an application that is difficult and time-consuming using medicinal chemistry techniques.

Acknowledgments.

The authors are grateful for support by grants from the National Institutes of Health (R01GM106137 and R01GM114237) and from the Cancer Prevention Research Institute of Texas grant (RP160657 and RP180880), and The Welch Foundation (F-1390)

Concluding Statements

This paper consists of a robust assessment of the capabilities of AMOEBA on GPU to enable drug discovery studies. The accuracy of prediction achieved (tau of 0.5, and R^2 0.75) are unlikely to be matched by other approaches. Studies on the capabilities of 5 different proteins to predict ligand affinity in the D3R grand challenge2 revealed that 3 of 5 protein targets had the best-submitted approach with a tau value that was worse than that observed in our studies[2]. Those approaches that scored better than a tau of 0.5 were universally machine learning tools. Therefore, these predictions explicitly included data in known protein-ligand pairs, and thus would not be usable in the case of authentic de novo ligand design.

The ability of the utilized approach to accurately predict relative ligand binding affinity is likely improved by the approach of explicitly fitting all of the IN17 scaffold torsions. Under current protocols, torsional assignment by poltype is accomplished via the reading in of torsional estimates from a torsional dictionary. These dictionary lookup values are only approximate; this approach ignores the effects of the surrounding atomic

environment. By specifically parameterizing the IN17 scaffold (and then automatically fitting any new torsions introduced in derivatization), we can ensure that all torsions closely match the quantum mechanics energy surface. This allows for capturing the effects of local ligand environment on torsional fit, which cannot be easily accounted for in a dictionary lookup-based approach.

In order to ensure this degree of torsional fit can be obtained in a rapid, high throughput manner, it would be preferable that torsional fitting be a fully automated process. Given the number of quantum calculations needed for torsional estimation (at least 6, 1 calculation for every 30 degrees), a procedure should be programmed that fragments the ligand and calculates each torsion. This fragmentation would allow for torsions to be determined in a relatively rapid manner, without the need to simulate parts of the ligand that have limited influence on the torsional parameterization.

While this study was successful at the prediction of relative binding free energy, several issues complicated the analysis, primarily as related to the absolute binding free energies. First, the possibility of a HEPES-E15 interaction was never conclusively excluded as a possibility. Experiments were done were the HEPES buffer was titrated out for Tris-HCl. However, it is still possible that both buffers are interacting at this site. A more conclusive study would consist of a series of titration down in buffer concentration and extrapolate to buffer free binding K_d. If HEPES is interfering with IN17 inhibitor binding, one would expect to see an increase in the apparent potency of IN17 with a reduction in buffer concentration. Another limitation in the presented study is the lack of an IN17-MELK co-crystal structure. Without this crystal structure, it is not possible to

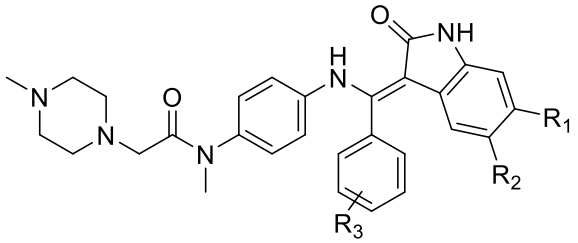
conclusively determine the isomerization state of the terminal carboxyl group. Therefore, for this study, we assumed that IN17 binds in the same conformation as observed in the structure of the IN17 solvent phase (non-protein bound) crystal. It is possible that multiple crystal forms of IN17 exist, and that some mixture of the two possible isomers is produced. A final concern is the protonation state of the piperazine group. It is highly likely that only one of the two piperazine nitrogen atoms is protonated. Based on computational evidence, I would hypothesize that this interaction is not critical for binding. However, this is not supported by the co-crystal of nintedanib and MELK, which has an E15-piperazine interaction. It is possible that this interaction is a crystallization artifact, especially given the surface exposed nature of the region. Mutagenesis studied may be able to determine if this interaction is essential to the inhibition of IN17.

Even without possible improvements in the accuracy of the AMOEBA forcefield, current relative ligand binding metrics are likely good enough to enable a new application - namely the design of selectivity into inhibitors. Human kinases have evolved through a series of gene duplication events[178]. Therefore, for any given target kinase, there are often multiple off-target kinases that are also inhibited by a target drug. This can be an advantage, as targeting multiple pathways with one drug can make the acquiring of resistance mutants challenging. For example, the anti-cancer drug sunitinib is known to target VEGFR1-3, PDGFRα and ß, c-kit, FLT3, CSF1R, and RET[50, 179]. Since these kinases are all anti-cancer targets, sunitinib may have anti-cancer activity through multiple different mechanisms, increasing potency. However, these off-target inhibitions can lead to toxicity. Using sunitinib as an example again, sunitinib is associated with

cardiotoxicity due to inhibition of AMPK. [180]. In drug development projects, it is common to investigate lead compounds with high throughput technologies which identity which kinases are inhibited by a compound[181]. One could then, in theory, use free energy computational tools to ensure that new derivatives hit desired kinases, but not off-target kinases. This negative design approach has not been extensively utilized in drug discovery, as it has not been feasible to predict binding vs. non-binding compounds accurately. This binding free energy approach has the accuracy to accomplish negative design via the testing of affinity to off-target kinases, an approach that could reduce the chances of off-target effects early in the drug discovery process.

Tables

TABLE 1: GROUPS PRESENT AT R_1, R_2, AND R_3 FOR THE DERIVATIVES TESTED.



Compound	R_1	R_2	R_3
IN17	H	$C(=O)OCH_3$	H
16	H	H	H
22	$C(=O)NH(CH_2)_3N(CH_3)_2$	H	H
23	$C(=O)NH(CH_2)_3NH_2$	H	H
25	H	$C(=O)N(CH_3)_2$	H
18a	H	$C(=O)OCH_3$	*p*-NO_2
18b	H	$C(=O)OCH_3$	*p*-NH_2
18d	H	$C(=O)OCH_3$	*p*-$C(=O)OCH_3$
18e	H	$C(=O)OCH_3$	*p*-OCH_3
18i	H	$C(=O)OCH_3$	*m*-, *p*-(1,3)-dioxol
18g	H	$C(=O)OCH_3$	*m*-OCH_3
18p	H	$C(=O)OCH_3$	*m*-NO_2

TABLE 2: COMPUTATIONAL PREDICTIONS AND EXPERIMENTAL BINDING ENERGIES (IN KCAL/MOL). ALL RELATIVE VALUES USE THE IN17 VALUE AS REFERENCE (0 KCAL/MOL). AS EXPLAINED IN THE MAIN TEXT, AN ADDITIONAL RESTRAINED SIMULATION WAS USED TO OBTAIN BINDING FREE ENERGY FOR COMPOUND 18G. TO WITHIN ONE DECIMAL PLACE, UNCERTAINTY FOR EACH OF THE RELATIVE PREDICTIONS IS 0.2 KCAL/MOL, AND UNCERTAINTY IN THE EXPERIMENTAL VALUES IS 0.1 KCAL/MOL. OPTIMAL RMSD IS 0.8 KCAL/MOL, AND RAW RMSD IS 1.1 KCAL/MOL.

	relative prediction	relative experimental
18a	2.2	0.6
18b	1.3	-0.6
18d	1.5	0.7
18e	-0.3	-0.1
18g	1.1	-0.5
18i	1.0	-0.1
18p	1.0	1.6
16	0.9	0.4
22	4.9	4.4
23	3.7	>4.7
25	2.6	2.2

Figures

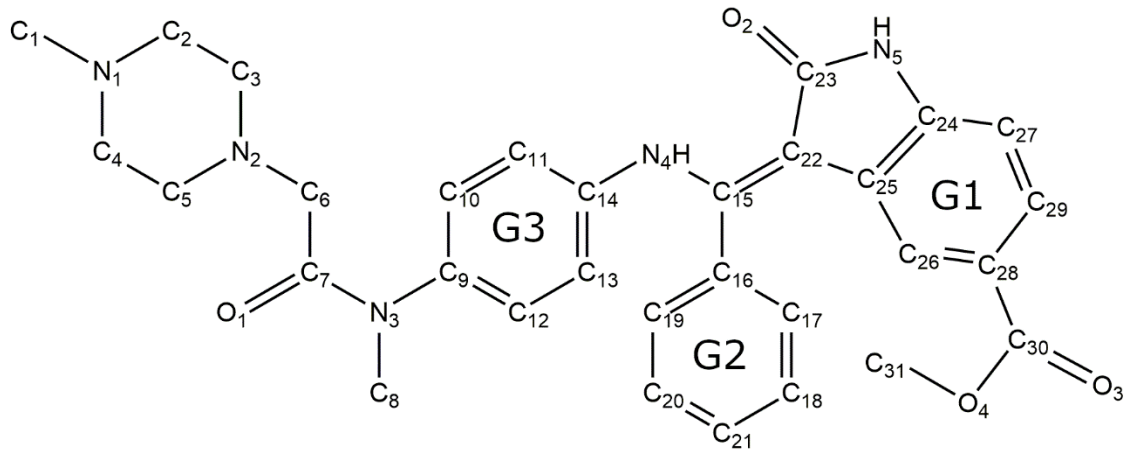


FIGURE 1: STRUCTURE OF IN17. ATOMIC LABELS AND RING GROUP NUMBERS ARE REFERRED TO THROUGHOUT THE PAPER.

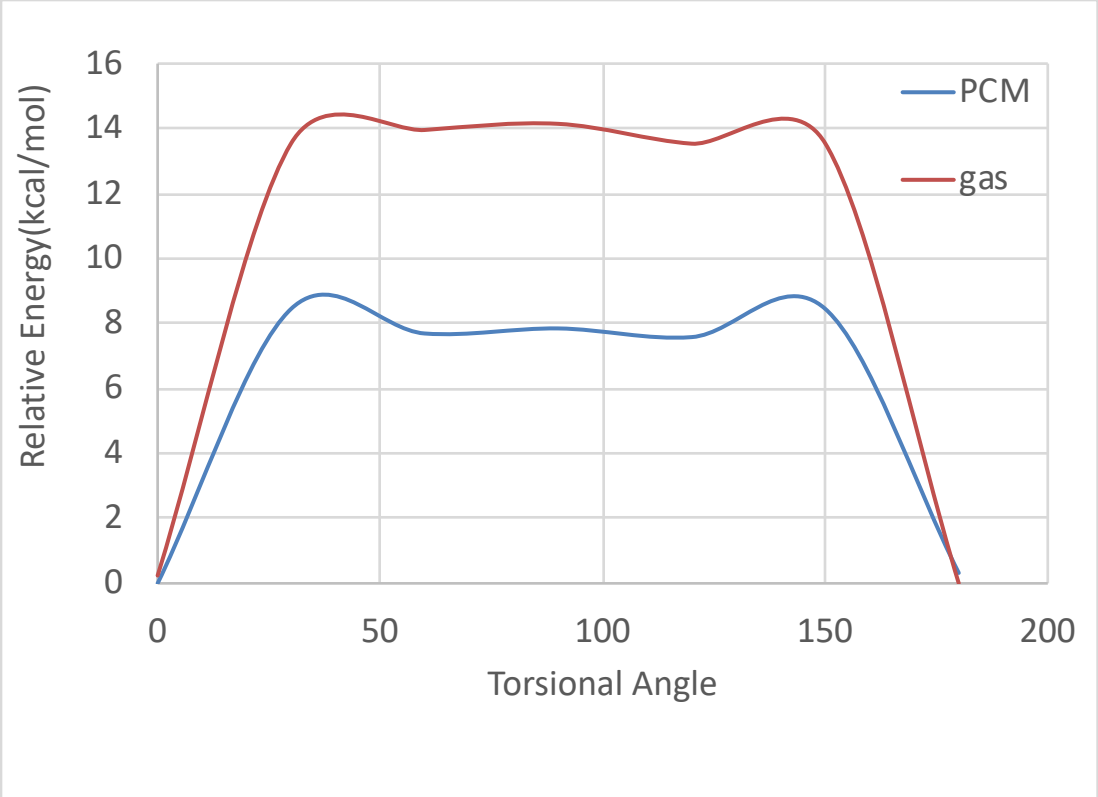


FIGURE 2: ROTATIONAL BARRIER FOR THE O3-C30-C28-C29 TORSION OF IN17. PCM (POLARIZABLE CONTINUUM METHOD[172]) IS USED TO CAPTURE THE SOLVENT EFFECT. ALL QM ENERGIES WERE CALCULATED USING MP2/6-311+G, WITH ROTATIONS AT EVERY 30 DEGREES.**

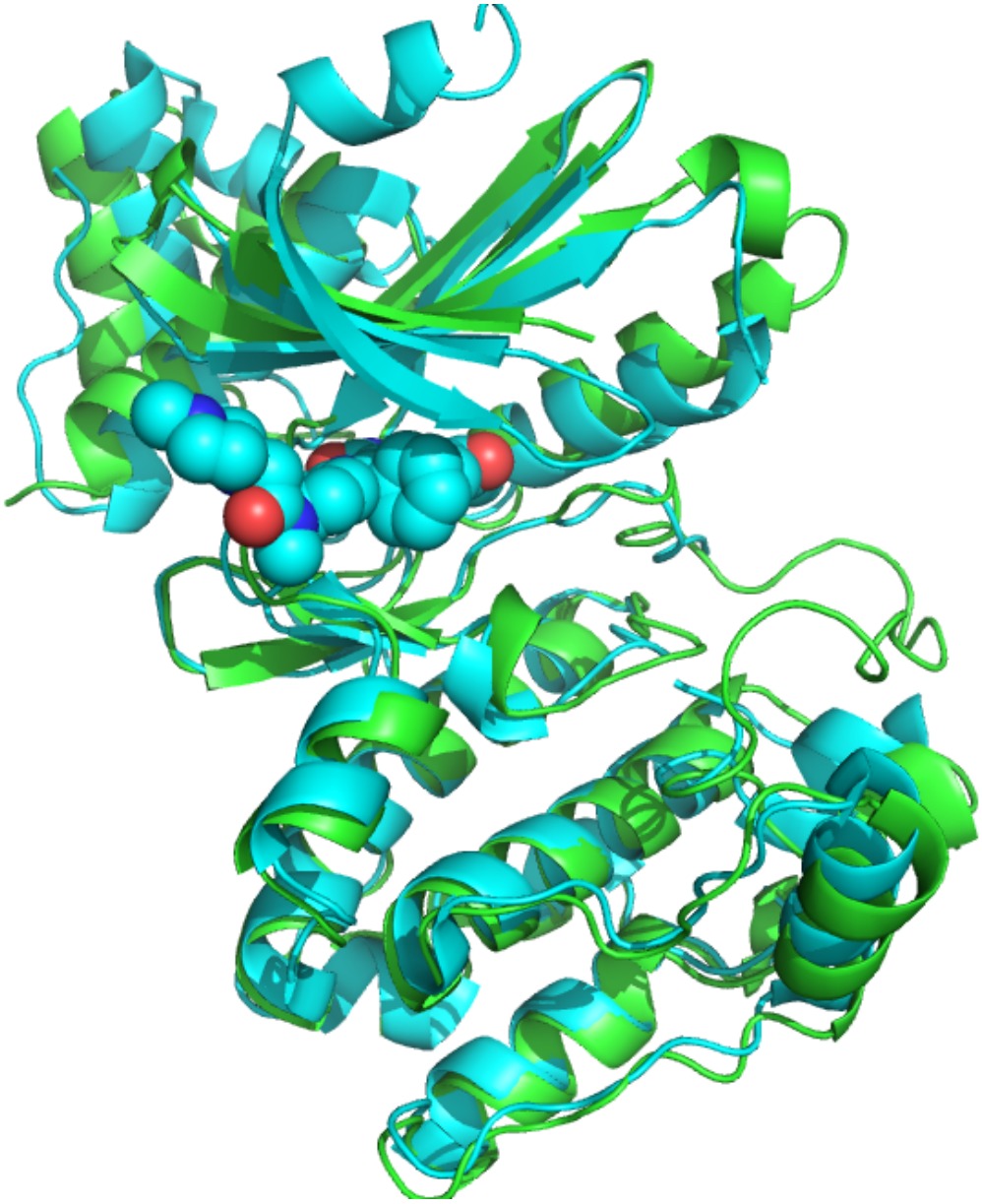


FIGURE 3A: SUPERPOSITION OF CRYSTAL STRUCTURE 5MAF (CYAN) AND SIMULATION ENDSTATE OF A MELK-NINTEDANIB SIMULATION OF 10NS (GREEN). FOR CLARITY, THE NINTEDANIB LIGAND FROM THE SIMULATION IS OMITTED.

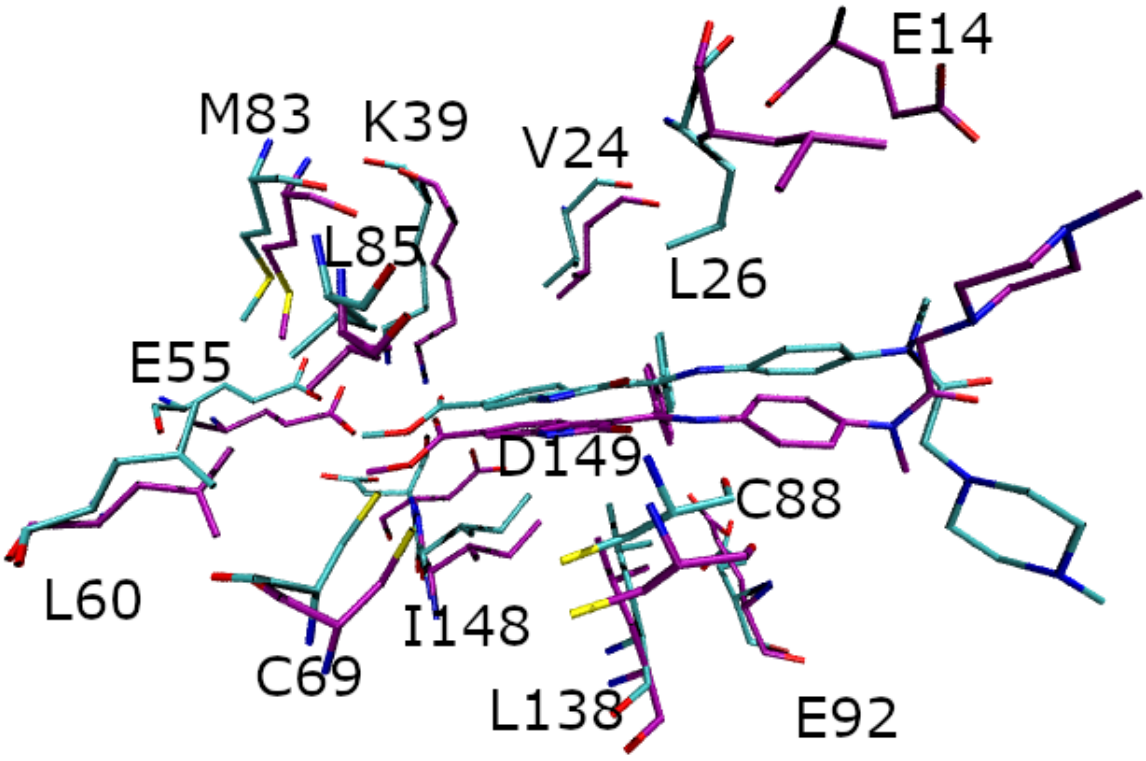


FIGURE 3B: COMPARISON OF THE BINDING SITE STRUCTURE OF THE SIMULATION OF NINTEDANIB-MELK (CYAN) AND THE 5MAF CRYSTAL STRUCTURE (PURPLE).

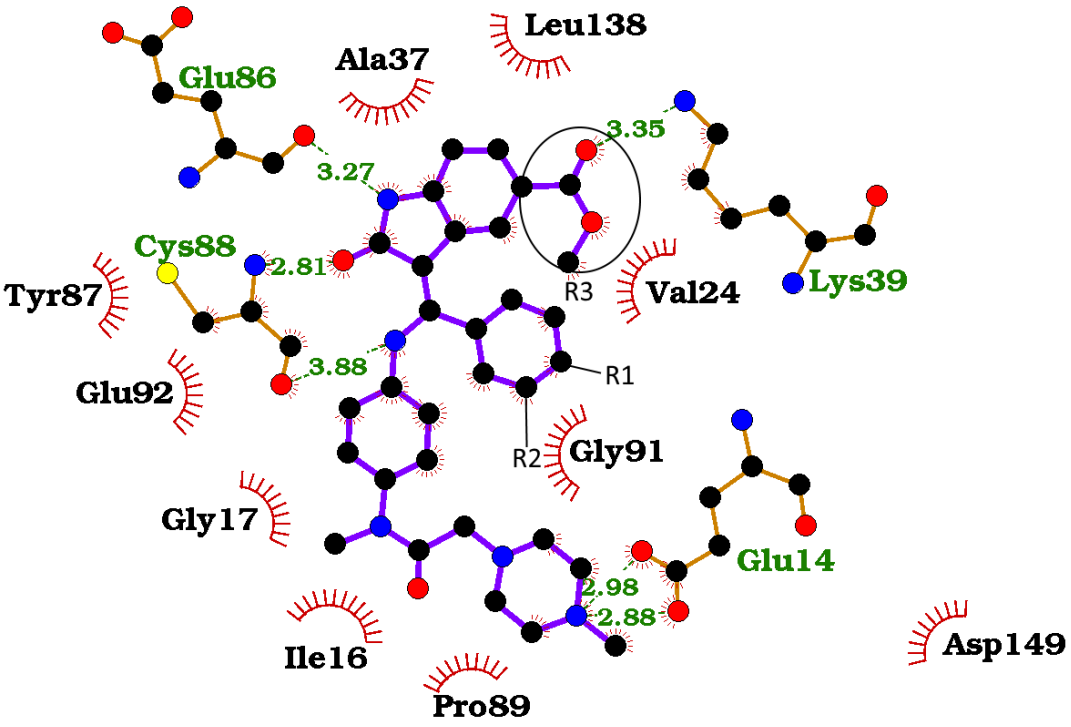


FIGURE 4: INTERACTION MAP OF IN17 MELK BINDING. THE POSITIONS OF SUBSTITUTION GROUPS R1, R2, AND R3, ARE LABELED. IMAGE GENERATED USING LIGPLOT+[182].

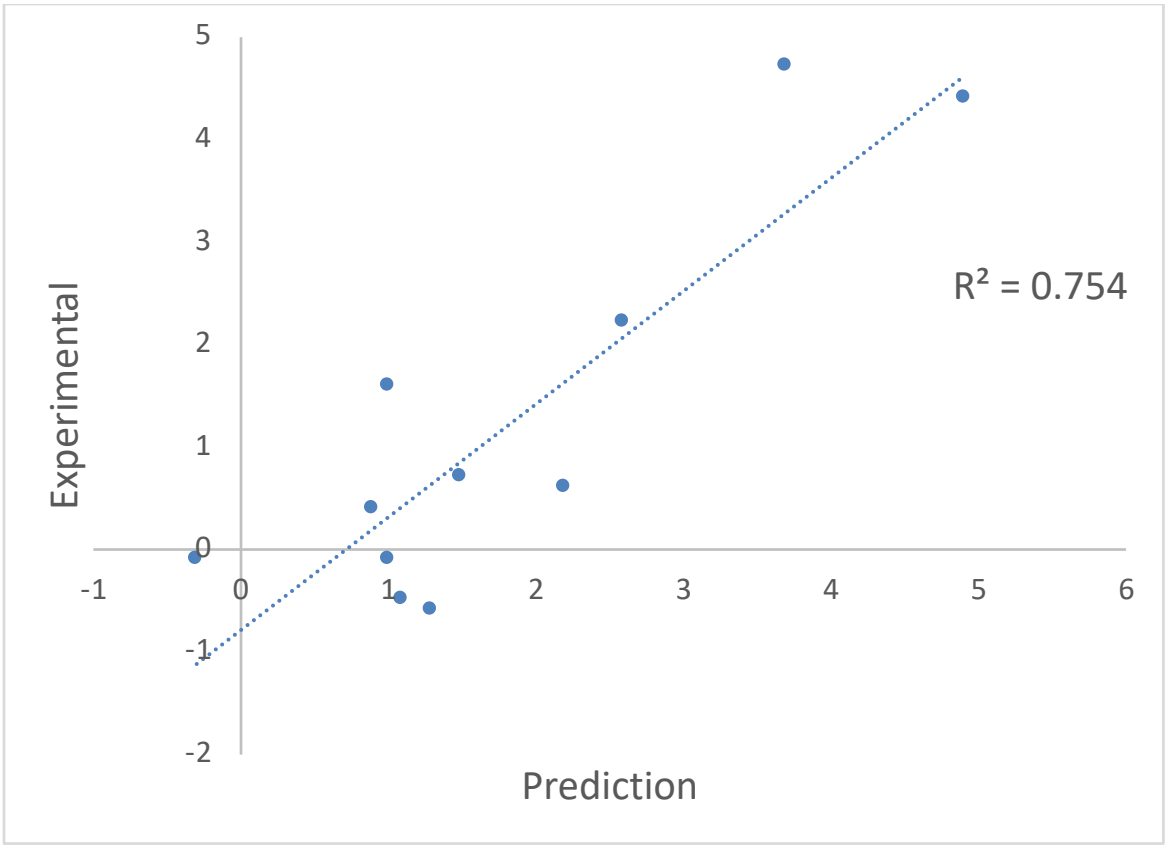


FIGURE 5: CORRELATION BETWEEN EXPERIMENTAL BINDING AFFINITY AND COMPUTATIONAL PREDICTION. ALL VALUES IN KCAL/MOL.

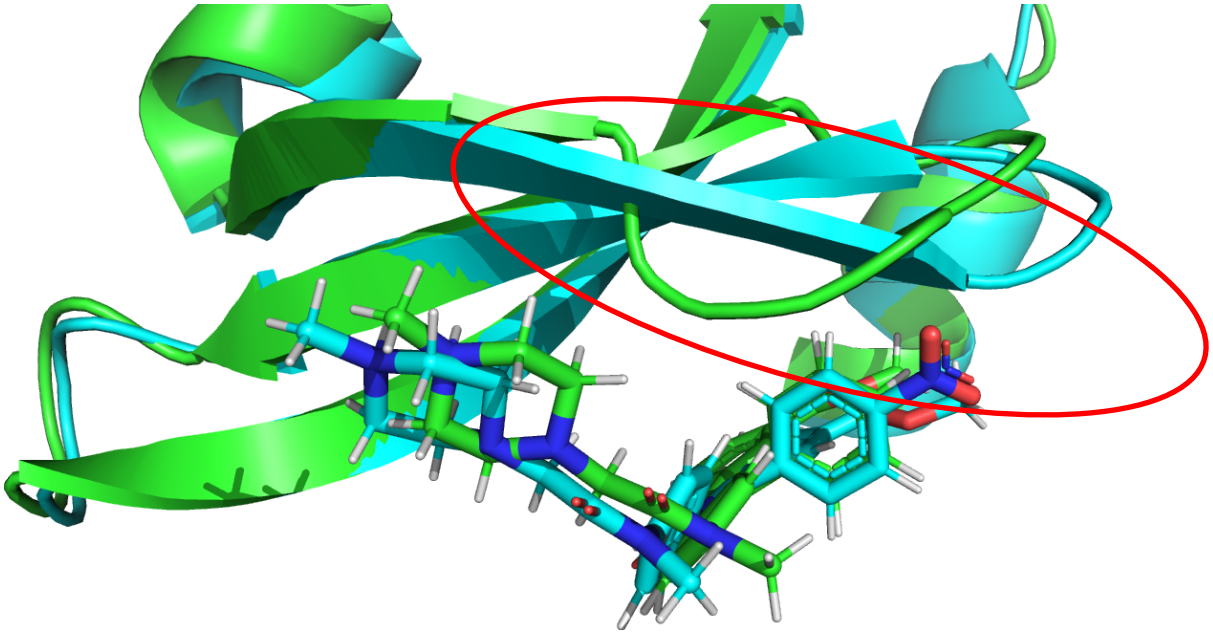


FIGURE 6: COMPARISON OF IN17 SIMULATION STRUCTURE (GREEN) AND 18A SIMULATION STRUCTURE(CYAN). ONLY THE FIRST 50 RESIDUES ARE SHOWN FOR CLARITY. THE LOOP STRUCTURE DISCUSSED IN THE MAIN TEXT IS ENCLOSED IN THE RED CIRCLE.

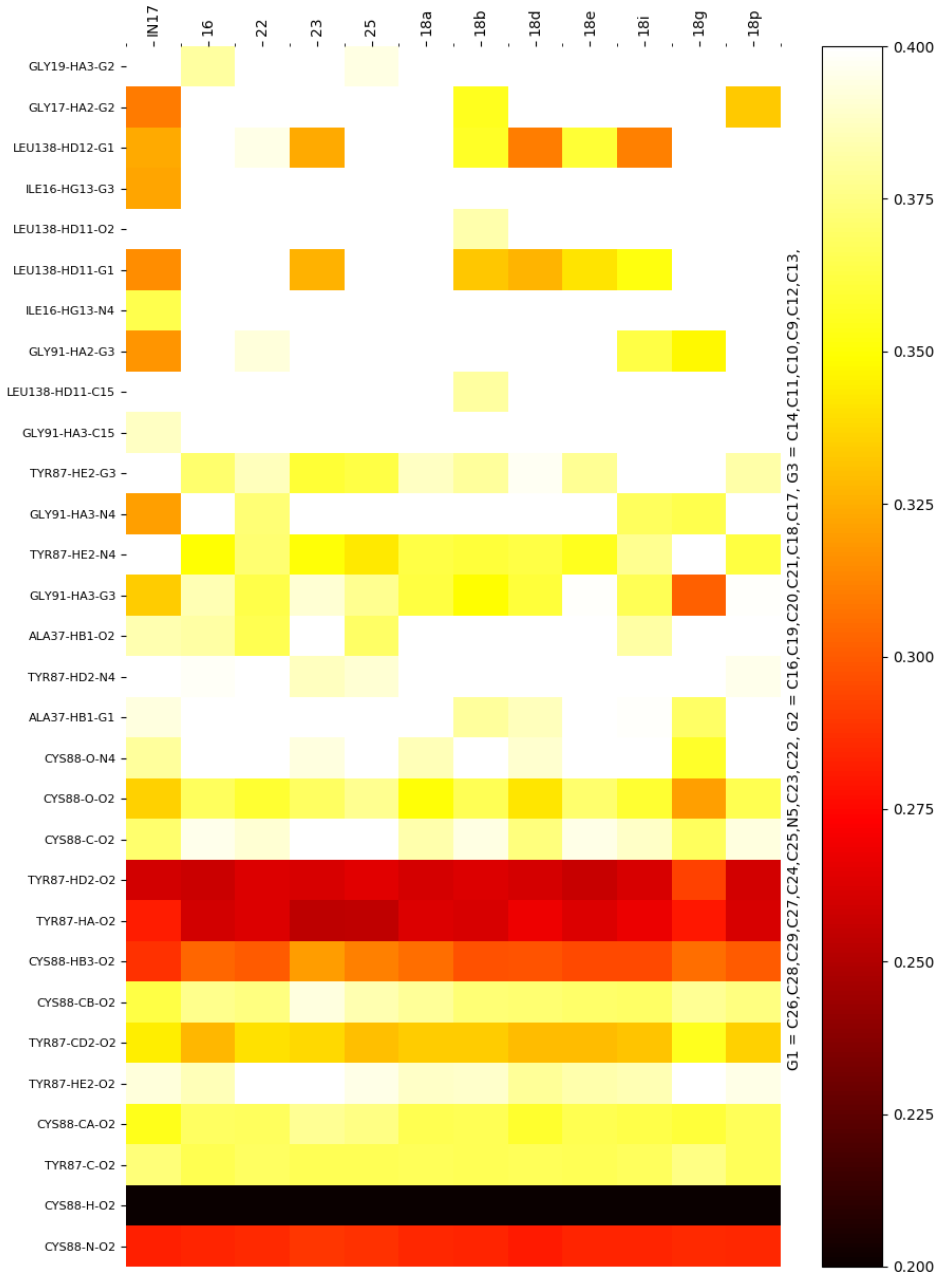


Figure 7: HEATMAP OF LIGAND-PROTEIN INTERACTIONS ACROSS ALL STUDIED LIGANDS. COLOR CORRESPONDS TO AVERAGE CONTACT DISTANCE (NM) ACROSS ALL 3NS OF MOLECULAR DYNAMICS SIMULATION. THE HEATMAP IS ORDERED BY MOST CONSERVED INTERACTIONS ACROSS ALL DERIVATIVES.

VIRIAL BASED BERENDSEN BAROSTAT ON GPUS USING AMOEBA IN TINKER-OPENMM[183]

Introductory Statements

This study chronicles the work on the inclusion of virial-based barostats into Tinker-OpenMM for polarizable multipole-based AMOEBA potential. Prior to this study, only the Monte Carlo barostat was implemented in Tinker-OpenMM. This acted as a non-trivial limitation for anyone wanting to code in the alternative, virial based barostats. The coding of the virial into Tinker-OpenMM varied in complexity depending on which term was being included. Some of the terms were coded into Tinker-OpenMM in a manner identical to base Tinker, with only minor syntax changes necessary. The virial for these terms was trivial to port over and consisted of merely copying and pasting of these terms to the end of the appropriate CUDA Kernel. Some terms, however, were quite challenging, most notably the electrostatic force. This force was broken up into many different components, most of which shared limited code terms with the Tinker CPU code. This required extensive reverse engineering, especially of the polarizable multipole PME related virial terms. The complete inclusion of the virial opens many possibilities with regards to further development of virial-based pressure control.

Introduction:

Proper pressure control is essential for molecular dynamics (MD) that requires simulation of pressure effects[39]. For example, molecular dynamics has proven invaluable in the prediction of the structure of compounds such as glasses[184], nanomaterials[185], and metals[186,187] under extreme pressures as high as 1,000,000 atm. Many of the standard spectroscopic techniques are ineffective under high pressures. For example, even pioneering NMR studies are limited to around 2000-3000 atm[188]. Since NMR is the

primary technique to gain dynamics information about molecular systems, MD simulations act as a valuable complement to the limited experimental tools available at higher pressures. As another biological example, proteins have evolved to maintain structure and function at the pressure experienced by an organism. For most organisms, this pressure is near 1 Atm. However, there has been increasing interest in the dynamics of proteins from piezophiles that live under extreme pressures as high at 1100 atm[189], pressures that would denature most proteins. Molecular dynamics studies of the pressure stability of these enzymes have revealed that protein dynamics of pressure tolerant enzymes (at least in Dihydrofolate Reductase) are altered to enable increased flexibility at high pressures[190], thus enabling substrate exchange. A better understanding of these adaptations may allow for biosynthetic applications or mutagenesis of proteins for high-pressure industrial applications[191]. Material science and biochemical studies such as these require robust pressure control that allows for simulation is not only ambient pressures, but also at extreme pressures. This pressure control is implemented via a simulation component known as a barostat.

Most barostat implementations require the calculation of a system property known as the virial[192]. The virial is defined as the change in energy with respect to volume (i.e., dU/dV). The virial is the sum of two components; an internal, potential interaction derived component and a kinetic energy term. For most systems, the internal virial has a tendency to push system volume inwards, while the kinetic term pulls system volume outwards. For simple pairwise forces, the internal virial expression is also equivalent to the dot product of force and distance. However, for some forces, internal virial

calculations derived from dU/dV are required (such as during the calculation of the virial due to multipole forces using Ewald summation[193]). The virial can then be converted into an instantaneous pressure using the equation $P_{inst} = \frac{1}{3*V} * (2 * KE - W)$, with KE being kinetic energy, and W is the average of the diagonal components of the internal virial tensor for the case of an isotropic barostat. This instantaneous pressure is then used to scale box dimensions and coordinates in order to bring the instantaneous pressure closer to the target external pressure. There is a wide range of virial based barostats, including the Nose Hoover barostat[40, 194-195], the Berendsen barostat[39], and the Langevin piston method[41]. Note that Berendsen does not give correct ensemble fluctuation, whereas it is very effective to equilibrate the system to target pressure.

Tinker-OpenMM[64] is a modified version of OpenMM[65, 196] designed for GPU computation containing many features that are not present in the main release of OpenMM. These additions include the latest modifications of the AMOEBA polarizable forcefield[114, 166] and the ability to perform free energy perturbation calculations. However, Tinker-OpenMM lacks in pressure control methodologies. Since Tinker-OpenMM currently doesn't compute the AMOEBA virial (unlike the base Tinker package[197-198] for CPU computation), Tinker GPU pressure scaling can only be accomplished via a Monte Carlo Barostat[199]. The Monte Carlo barostat uses the target pressure and system energy to probabilistically select increases or decreases in system box size. The Monte Carlo barostat results in the correct equilibrium system density and volume ensemble. However, the Monte Carlo barostat is less effective than virial-based barostats in the equilibration of molecular systems that are far from equilibrium density

or for systems undergo substantial volume changes as a result of state changes (e.g., protein folding and unfolding). Therefore, it is desirable for Tinker-OpenMM to contain virial-based barostats.

When making an initial choice of barostats, we wanted to implement a barostat that was already present in Tinker CPU so as to enable comparisons of these already well-validated platforms with the GPU results. Tinker CPU currently supports the Nose-Hoover barostat and the Berendsen barostat. The Nose-Hoover barostat was initially considered. However, the Tinker implementation of the Nose-Hoover barostat contains a thermostat coupled to a barostat. This coupling introduces complications in analysis, leaving the possibility that any error is due to the thermostat, not just the barostat. By implementing the Berendsen barostat, temperature control can be handled by the already implemented Bussi thermostat[42]. One disadvantage of the Berendsen barostat is that correct volume ensembles are not achieved.[199] This makes the Berendsen barostat inappropriate for production simulations, and limits effective use cases to initial equilibration. However, implementation of the Berendsen barostat can act as an initial test of if a GPU based virial can enable consistent pressure control. This is especially important since forces (and thus the virial, which uses many force intermediate terms in its calculation) needs to be calculated at a lesser, 32-bit precision (as opposed to the 64-bit computation utilized by Tinker CPU). This lack of precision could result in a drift in equilibrium densities in the GPU when compared to the CPU. Indeed, significant total energy drifts are observed when integrating positions at 32-bit precision, indicating that calculation of system properties using 32-bit precision can cause a significant difference

in system behavior[196]. Thus typical GPU MD utilized mixed precisions where certain variables such as positions use 64-bit while others use 32. Since the AMOEBA internal virial (or indeed, any polarizable virial) has not been implemented in CUDA for GPU-MD, it is unclear if a GPU based implementation of the AMOEBA virial would enable sufficient precision for robust pressure control. In this paper, we show the implementation of the AMOEBA virial on GPUs within Tinker-OpenMM and use the Berendsen Barostat as a test of the utilization of this virial for pressure control schemes.

Methods:

DERIVATION OF VIRIAL:

Given unit cell vectors, $\underline{a}_\alpha = [a_{\alpha 1}, a_{\alpha 2}, a_{\alpha 3}]^T, \alpha = 1, 2, 3$, which form the edges of the cell, the cell-matrix is defined as

$$\boldsymbol{A} = [\underline{a}_1, \underline{a}_2, \underline{a}_3] = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \qquad (1)$$

The instantaneous pressure is given by [200]

$$p_{\alpha\beta} = \frac{1}{V}\left(\sum_i m v_{i\alpha} v_{i\beta} - W_{\alpha\beta}\right) \qquad (2)$$

The first term on the RHS corresponds to the kinetic energy contribution while the second term is the internal virial:

$$W_{\alpha\beta} = -\sum_{i,j>i} r_{ij,\alpha} f_{ij,\beta}$$

The average of three diagonal components of pressure tensor gives the usual scalar pressure in an isotropic system.

The virial tensor can also be evaluated from derivatives of potential energy [201]

$$\frac{\partial U}{\partial a_{\alpha\beta}} = \sum_{\gamma=1}^{3} W_{\alpha\gamma}\, a_{\beta\gamma}^{-1} = V \sum_{\gamma=1}^{3} \prod_{\alpha\gamma} a_{\beta\gamma}^{-1} \quad (3)$$

which can be conveniently applied to obtain virial for a system handled by Ewald sum

$$W_{\alpha\beta} = \sum_{\gamma=1}^{3} \frac{\partial U_{Ewald}}{\partial a_{\alpha\gamma}} a_{\gamma\beta} \qquad (4)$$

Note the partial derivative is with respect fixed *s*, fractional coordinate.

The Ewald energy for multipoles is

$$U_{Ewald} = U_{real} + U_{recip} + U_{self} \qquad (5)$$

The self-energy term is independent of cell dimension and makes no contribution to pressure. The real space component of pressure tensor intuitively takes the form [202]

$$W_{\alpha\beta}^{real} = -\sum_{i,j>i} r_{ij,\alpha}\, f_{ij,\beta}^{real}$$

where $f_{ij,\beta}^{real}$ is the force between site *i* and *j* computed in the Ewald real space, and the summation is over all pairs of sites. The remaining reciprocal component is to be derived using above eq (4) and is presented in the supplementary materials of the publication.

The virial due to torque is calculated as $\Delta X_x * Fx_x + \Delta X_y * Fx_y + \Delta X_z * Fx_z$. Subscripts denote the X, Y, and Z frame defining atoms. All other terms are calculated analogously. This dot product definition of the virial can be used because this torque is not volume dependent[203]. Here forces are the forces converted from/due to torque.

VIRIAL IMPLEMENTATION:

Calculation of the diagonal and off-diagonal components of the virial tensor was added to the end of the appropriate GPU force kernels (contained in Tinker-Openmm/plugins/amoeba/platform/cuda/src/kernels) via adaptation of the vir() array

modifications present in the Tinker CPU codebase. Modifications were made to the Multipole, van der Waals, angle, angle-torsion, bond, out of plane bend, pi-torsion, stretch-bend, stretch-torsion, torsion-torsion, and torsion forces. For all forces except the more complicated multipole force (along with polarization), all changes were made at the end of the kernel, guarded by an if USES_VIRIAL preprocessor directive. The multipole virial is contained throughout multipole. cu, multipolePme.cu, and pmeMultipoleElectrostatics.cu, with all virial components, either utilizing the USES_VIRIAL directive or are contained within routines that are executed only if the virial is required for a given simulation. The flagging of virial-requiring terms with the USES_VIRIAL directive enables virial-dependent calculations only in simulations in which the virial is required, removing computational expenses in simulations that do not use the virial. GPU virial computation can be turned on in the interface by a call to OpenMM_System_setUsesVirial (omm->system, (OpenMM_Boolean) true). The GPU virial was split into fast, bonded (available by calling CudaContext.getFastVirial()) and slow, nonbonded (available by calling CudaContext.getSlowVirial()) components. This separation allows for the implementation of multistep algorithms (such as r-RESPA[204]) that require averaging of fast components by the number of inner steps.

BERENDSEN BAROSTAT IMPLEMENTATION:

The Berendsen barostat was coded for use with the RESPA integrator present in Tinker-OpenMM. Briefly, a routine scaleBox() was added into the CustomStepKernel implementation present in platforms/cuda/src/CudaKernels.cpp. The scaleBox routine

performs pressure scaling identical to that present in the Tinker interface, based on the barostat scheme reported previously[39]. Briefly, after each step, the kinetic energy and virial potential were used to determine the box length scaling constant according to the equation $Lengthscale = ((1 + \Delta t * \frac{\beta}{\tau} * (P_{\text{inst}} - P_{\text{target}}))^{1/3}$. The compressibility ($\beta$) was chosen as that of water (0.000046), with τ equaling the default Tinker CPU value of 2.0. The instantaneous pressure was calculated using the equation $P_{inst} = \frac{k}{3*V} * (2 * KE - W)$, with k being a conversion factor between kJ/mol/Å^3 to atm (equal to 16.39). In order to increase system stability, the fast virial was averaged across all inner steps. The actual scaling of atomic coordinates is accomplished using the GPU kernel scaleCoordinates() inside platforms/cuda/src/kernel/monteCarloBarostat.cu, which scales the positions of molecular centers (as opposed to scaling each individual atom independently). The command addscalebox () was placed at the end of the RESPA definition in the Tinker ommstuff.cpp interface, after the BUSSI, scaling routine, causing the scaleBox() routine to be called at the end of each r-RESPA step.

VIRIAL VALUE CONFIRMATION:

The virial build of Tinker-OpenMM was modified to print both the fast, bonded virial and the slow, non-bonded virial at each r-RESPA inner step. 1000 steps of MD were then performed using a 1.0 fs time step, with the output of a structural archive file every step. The total virial for each GPU generated frame was then calculated using the Tinker "analyze" routine. The average and percent difference was then calculated on a

per-component basis using only the diagonal component of the virial. The first 500 frames were ignored, in order to compare near-equilibrium frames

MD PROCEDURES:

All Molecular Dynamics calculations were performed using a 2.0fs timestep, and the RESPA integrator, with structural output every 1ps. All calculations were performed at 1Atm pressure using the Berendsen barostat, and 298K temperature using the BUSSI thermostat[42] unless otherwise noted. Simulations utilized an electrostatic Ewald cutoff of 7.0 Å and a van der Waals (vdW) cutoff of 12.0 Å.

MOLECULAR SYSTEMS:

All small-molecule systems consisted of pure liquids available in the example/ folder of the tinker release and utilized the amoeba09 parameters. The water system utilized consisted of a cubic box of 2,210 water molecules run using the AMOEBA water14 forcefield. The protein system utilized was bench7.xyz solvated Dihydrofolate Reductase (DHFR) test system includes in the bench/ folder of the Tinker CPU distribution, using the amoebabio09 parameters. The RNA system utilized was a solvated double-stranded RNA molecule consisting of the sequence 5'-AAGCUGCCAG-3', 3'-UCGACGGU-5', using the amoebanuc17 parameter file.

Results:

VIRIAL CPU VS GPU COMPARISON

Since the virial for each force often contains mathematical intermediates of the force (or even the calculated force itself), a computationally efficient internal virial must be calculated at 32-bit precision. This potentially limits internal virial accuracy when compared to the 64-bit precision utilized by Tinker CPU. Initial tests of the internal virial consisted of calculating the difference in the diagonal internal virial components calculated using the Tinker-OpenMM GPU and Tinker CPU platforms. Since only the diagonal components contribute to isotropic pressure, the off-diagonal internal virial tensor components were ignored. This comparison was accomplished by performing 1000 steps of MD on a minimized starting structure, using a build of Tinker-OpenMM modified to print out the slow and fast virials every r-RESPA inner step. The first 500 structures of this simulation were not included in the internal virial analysis in order to test near-equilibrium values. The internal virial tensor for these final 500 frames was then calculated using Tinker's "analyze" routine. The Dihydrofolate Reductase (DHFR) protein system showed an average diagonal internal virial component difference of 9.0±6.0 kcal/mol/Å^3, with a percent difference of 0.08±0.06% (**Table 1**). The RNA system displayed slightly more significant raw divergence, at 14.3±9.6 kcal/mol/Å^3 due to larger system size. For the RNA system, the percent difference was identical to that of the protein system, with a percent difference of 0.08±0.05%. It was unclear if this degree of accuracy was sufficient to enable virial-based pressure control. Most of the divergence in CPU and GPU calculated internal virial was identified as being a result of the particle

mesh Ewald virial, which cannot be improved upon without increasing to 64-bit precision. The calculation at this increased precision would result in an unacceptable approximately 30-fold reduction in performance[196]. At this speed reduction, the performance advantage of the GPU platform is essentially negated. It was thus decided to proceed with Berendsen pressure control testing with this virial divergence.

EQUILIBRATION OF SMALL MOLECULE SYSTEMS:

Since it was uncertain if the internal virial accuracy was sufficient to enable pressure control, it was necessary to test the capabilities of the GPU platform on a wide range of small molecules. In addition to water, pure liquids of formamide, benzene, and methanol were chosen to represent a diverse set of molecular properties. This series of compounds was simulated at 1 Atmosphere pressure using Tinker CPU for 1ns to generate equilibrium structures and velocities using a well-validated computational platform. These simulation starting points were then run using the Berendsen barostat for 30ns at 1A tm to confirm that this equilibrium is maintained. All of the tested small molecules maintained the same density as in the CPU simulation (**Table 2**), an indication that the Tinker-OpenMM system is able to maintain stable simulations for a wide range of chemical moieties.

EQUILIBRATION OF WATER AT HIGH PRESSURES

It is essential that any barostat be able to equilibrate systems that are at an initial non-equilibrium density. The previous series of tests started the simulated systems at equilibrium and did not determine the ability of the Tinker-OpenMM platform to

simulate systems at pressures other than 1 atm. In order to test both capabilities, a series of simulations of water were conducted at 1, 1000, 2000, and 4000 atmospheres. Critically, all 4 series of simulations were started with structures and velocities of a 1 atm water system generated using CPU. Therefore, reaching the correct equilibrium requires that the higher-pressure systems increase in density, demonstrating the capability of the GPU Berendsen barostat to equilibrate to different pressures. This series of Berendsen simulations reached near-equilibrium densities within 20ps and displayed equilibrium densities virtually identical to those observed in the 1ns CPU simulation (Figure 1) that were started at 1 atm pressure. The minor shift observed in equilibrium densities is likely due to integration and velocity precision differences between GPU and CPU. The percent virial divergence is small enough so as this consistent shift is not easily explained by virial divergence. The previous test (with all compounds at 1 Atm) could have been passed by a barostat with little to no box size evolution. The change of box size to reach equilibrium densities occurs relatively rapidly, while still maintaining an appropriate long-term equilibrium. This is a strong indication that the Tinker-OpenMM Berendsen barostat can perform pressure equilibration to densities identical to that of the CPU Berendsen platform, despite the less accurate virial.

COMPARISON OF BERENDSEN AND MONTE CARLO BAROSTATS ON GPU

The only previous pressure equilibration platform present in Tinker-OpenMM was the Monte Carlo Barostat. Unlike the Berendsen barostat, the Monte Carlo barostat displayed correct ensemble pressure and velocity fluctuations and thus is more suitable

for production simulation. The weakness of the Monte Carlo barostat, however, is the equilibration of structures far from equilibrium or dealing with substantial volume changes. Therefore, a likely pipeline would consist of an initial equilibration with the Berendsen barostat, followed by production simulations using Monte Carlo, or another (to be developed) virial based barostat such as Nose-Hoover. In order for this pipeline to work effectively, the equilibrium densities of the Monte Carlo and Berendsen barostat should be in agreement. To test this agreement, the water density tests at 1, 1000, 2000, and 4000 atm, as well as the other liquids at 1 Atmosphere, were repeated using the Monte Carlo barostat in Tinker-OpenMM on GPUs. The GPU Monte Carlo barostat showed comparable densities to the GPU Berendsen barostat for both water (**Figure 2**) and organic liquids (**Table 3**). This close agreement enables a smooth transition between the Monte Carlo and Berendsen barostats. This agreement is even closer than that observed between the GPU and CPU Berendsen barostats. This is an indication that the small divergence observed between the two platforms may be due to precision issues related to the velocity and components of the platform rather than the barostat, and more importantly, the polarizable multipole virial has been correctly implemented on the GPU platform.

Conclusions:

Over time, Tinker-OpenMM is nearing the molecular dynamics capabilities of the Tinker-CPU platform. One of the most significant limitations of the Tinker-OpenMM platform has previously been the lack of virial-based pressure control methods. These

virial-based methods are often more stable during initial equilibration than the Monte Carlo barostat. While the Berendsen barostat lacks the proper ensemble, it is an essential steppingstone in virial-based pressure control on GPU. Prior to this study, it was unclear if the lesser accuracy of OpenMM force (and thus virial) calculation would be accurate enough to enable robust pressure control. The results of this study indicate that this inaccuracy is unlikely to be an issue in the implementation of pressure control schemes. In the near future, we aim to add a wide range of the diversity of barostats that require the virial, such as Nose-Hoover or the Langevin Piston.

Concluding Remarks

When we had initially set out to add virial-based pressure control, the Berendsen barostat was not our primary target pressure control scheme. When initial prototype results for this barostat was presented to the Tinker community, the reception was lukewarm. This was due to the fact that the Berendsen barostat does not result in the correct ensemble volume fluctuations. However, attempts to implement other barostats in the allotted time were unsuccessful. For example, attempts were made to code in Tinker-CPUs' Nose Hoover barostat. Initial results for this barostat looked promising- it appeared to maintain the density of water at the correct value. However, when inputted with alternative liquid boxes- such as methanol, biases to high density were observed. We were unable to accurately identify the causes of this errant behavior and decided to change focus to the Berendsen barostat (which was already coded) due to concerns with the timing of the writing of this thesis. However, in finalizing the Berendsen barostat, a

possible (untested) cause of the errant Nose-hoover behavior may have been identified. The Tinker-OpenMM getKineticEnergy() routine shifts atomic velocities by half a timestep of acceleration to account for the Verlet based shift of calculation of velocities and positions. This shift is not needed in the Nose-Hoover implementation but may be causing inappropriate kinetic energy to be calculated. While this has yet to be formally tested, this is a likely stepping off point to attempt to fix the Tinker-OpenMM Nose Hoover implementation.

Acknowledgments:

The authors are grateful for support by the National Institutes of Health (R01GM106137 and R01GM114237)

Tables:

TABLE 1:

THE AVERAGE AND ABSOLUTE VIRIAL DIVERGENCE BETWEEN SIMULATION FRAMES GENERATED USING BERENDSEN BAROSTAT MOLECULAR DYNAMICS (MD) ON GPU AND THE CPU ANALYSIS OF THESE FRAMES. GPU MD WAS RUN FOR 1000 STEPS OF MD USING A 1FS TIMESTEP, AND PER-COMPONENT AVERAGE DIVERGENCE WAS CALCULATED FOR FRAMES 500 TO 1000.

	Protein	RNA
Absolute Difference(kcal/mol/Å^3)	9.0±6.0	14.3±9.6
Percent Difference	0.08±0.06%	0.08±0.05%

TABLE 2 COMPARISON OF THE EQUILIBRIUM DENSITY OF CPU AND GPU BERENDSEN BAROSTAT SIMULATIONS. GPU RESULTS ARE TAKEN OVER 30NS, AND CPU RESULTS ARE TAKEN OVER 1NS. BOTH RESULTS IGNORE THE FIRST 200PS.

Compound	GPU Density	CPU Density
Benzene	0.878±0.007	0.877±0.007
Formamide	1.124±0.004	1.124±0.004
Methanol	0.781±0.002	0.783 ±0.002
Water	0.994±0.003	1.002±0.003

TABLE3: COMPARISON OF EQUILIBRIUM DENSITY (OVER 30NS, IGNORING THE FIRST 200PS) FOR THE BERENDSEN AND MONTE CARLO GPU BAROSTATS.

Compound	Berendsen	Monte Carlo
Benzene	0.878±0.007	0.880±0.009
Formamide	1.124±0.004	1.124±0.006
Methanol	0.781±0.002	0.782±0.004
Water	0.994±0.003	0.998±0.005

Figures

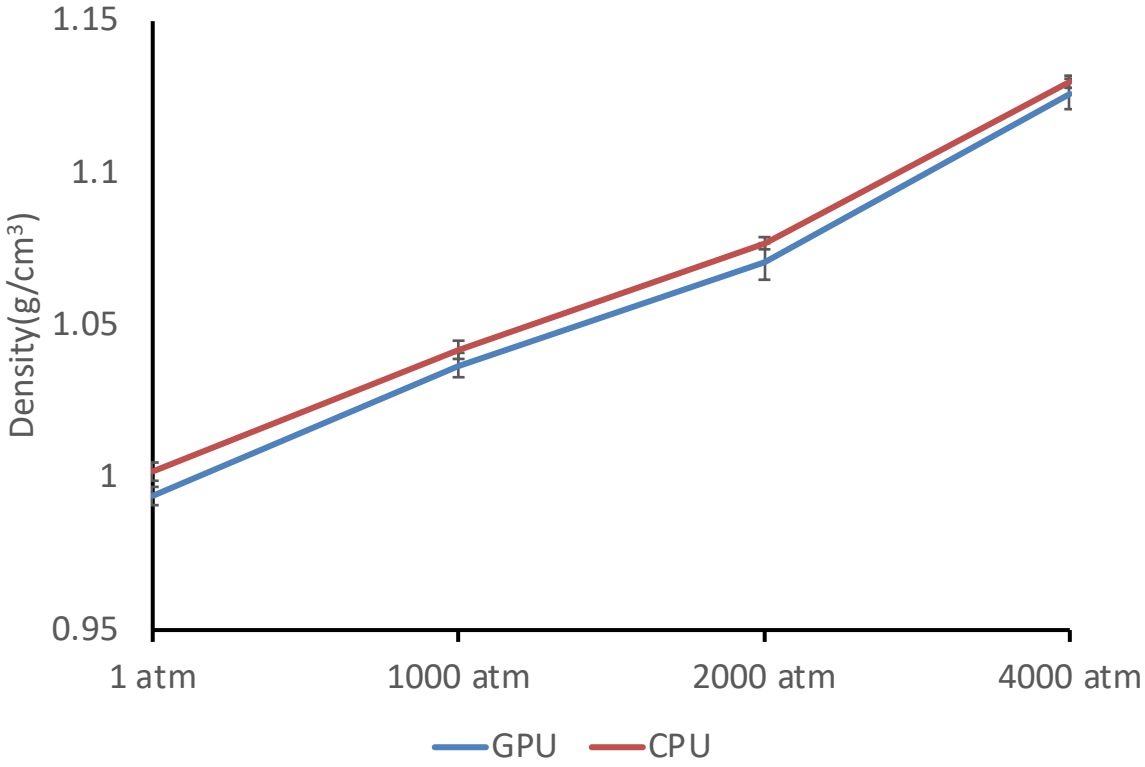


FIGURE 1: AVERAGE DENSITY FOR BERENDSEN GPU AND BERENDSEN CPU MD SIMULATIONS ON THE WATER AT VARIOUS PRESSURES. VALUES ARE REPORTED AS AVERAGE ± STANDARD DEVIATION. FIRST 200PS OF SIMULATION TIME WAS IGNORED IN THE CALCULATION. CPU SIMULATIONS WERE CONDUCTED FOR 1NS, AND GPU SIMULATIONS WERE SIMULATED FOR 30NS.

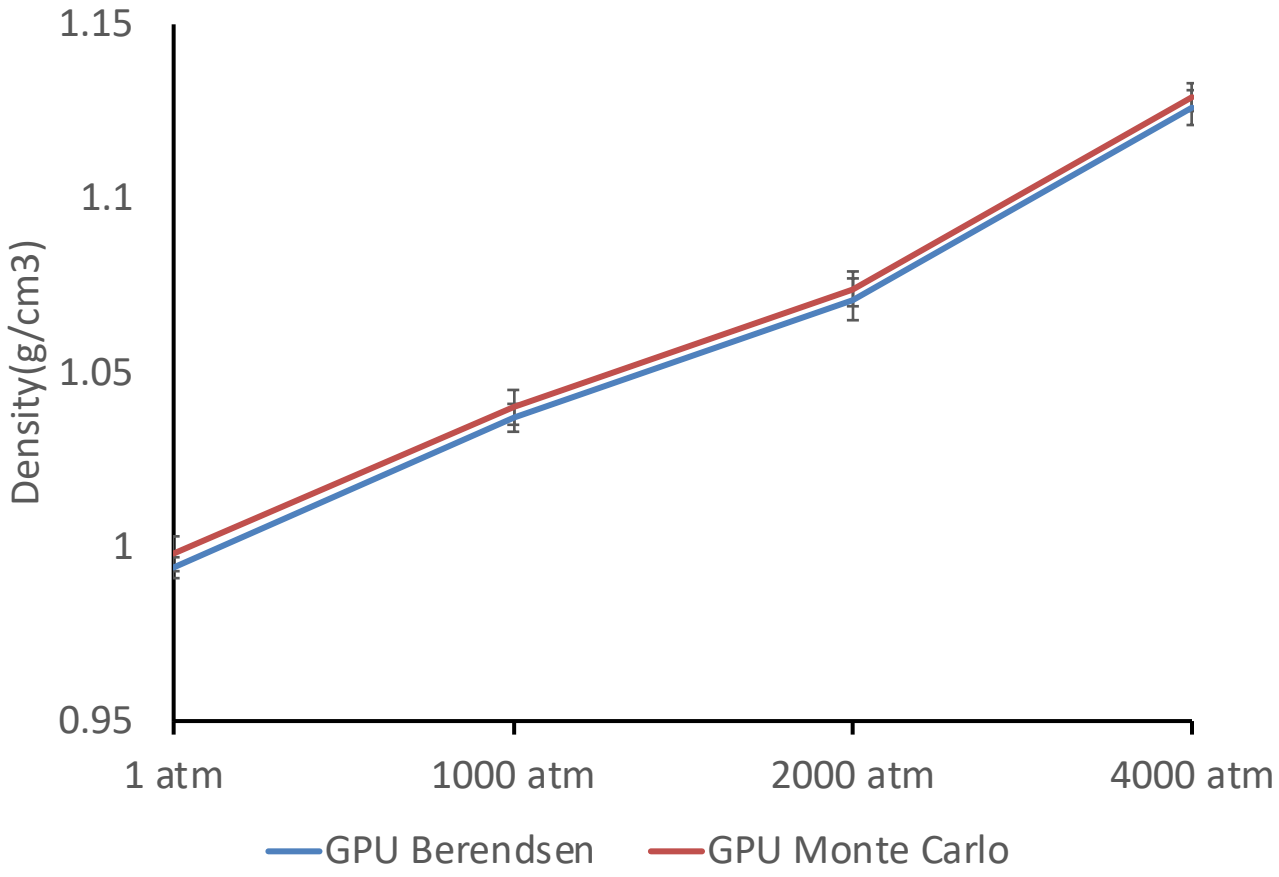


FIGURE 2: AVERAGE DENSITY FOR BERENDSEN GPU AND MONTE CARLO GPU SIMULATIONS ON THE WATER AT VARIOUS PRESSURES. VALUES ARE REPORTED AS AVERAGE ±STANDARD DEVIATION. FIRST 200PS OF SIMULATION TIME WAS IGNORED IN THE CALCULATION. BOTH SERIES OF SIMULATIONS WERE SIMULATED FOR 30NS.

FUTURE DIRECTIONS

During my time as a graduate student, the usability of Tinker-OpenMM as a GPU molecular dynamics engine has improved dramatically. During the early stages of my Ph.D., simple NVT simulation (or constant pressure simulation using Monte Carlo) was the only functionality of AMOEBA on Tinker-OpenMM. Through the work of myself and others, Tinker-OpenMM has reached a point of maturity where it is mostly ready for wide-scale adoption for various free energy calculations. Now that we have reached this degree of implementation, the question comes, what is needed in the next stages of the evolution of this platform, as well as the use of AMOEBA as a whole?

The first, most noticeable improvement is in the functional form of the AMOEBA forcefield. Development of a next generation of the AMOEBA forcefield (AMOEBA+) is well underway[205]. This update promises to allow for greater accuracy in results due to a better capturing of the physics of the electrostatic force. The most noticeable changes in this update to the AMOEBA forcefield is the addition of charge transfer and charge penetration terms. Initial analysis of the AMOEBA+ model on the water is promising, and parameterization of a protein forcefield using these new terms is ongoing. This update is (hopefully) the last major update to the functional form of AMOEBA. Having stability in the overall functional form of AMOEBA should aid in needed developments in other areas of the utilizing of Tinker OpenMM.

The next most urgent area that needs significant improvements is in the area of general usability. The computational chemistry community as a whole is impressed by the capabilities of the AMOEBA forcefield and want to integrate AMOEBA based

calculations into their workflows. However, the usability of Tinker leaves much to be desired with respect to the new user experience. Where most comparable packages such as AMBER and CHARMM have GUIs and extensive tutorials, Tinker solely has unix command line programs and lesser documentation. The addition of a GUI based pipeline for (for example) binding free energy simulations would greatly ease the acquisition of new users of this powerful forcefield.

Another problematic area is that of small molecule parameterization. It is clear that the parameterization process can work effectively (as evidenced by my work with MELK ligands). However, the process (especially for torsional optimization) is not fully automated and requires a large number of expensive QM calculations for torsional scanning. One solution to the problem of torsional parameterization is that of a torsional lookup dictionary. However, it is likely that torsional values are too dependent on neighboring atomic environment to be adequately captured by this approach. A better solution would be to "slice" a ligand into smaller fragments, each of which would contain a single torsion and its environment. This would allow for QM torsional scans that can be completed in a reasonable amount of computational time. Such a program would be challenging to implement since this would require some molecular insight into what constitutes a viable fragment (as a simple example, one should not split up a ring system). However, this approach to ligand parameterization is the most likely to be able to generate useful molecular parameter sets. This approach of performing extensive QM calculations for each new molecule is expensive, but not unprecedented. Indeed, this approach is precisely how multipoles are calculated for new molecules. If such an

approach could be made computationally feasible, it would likely result in increases in usability and accuracy.

Another area of improvement for Tinker-OpenMM is in the ability to speed up simulation speed. One example of such efforts includes the recent addition of OPT3 polarization[206], a truncation to the number of polarization iterations that results in a 30% overall improvement in system performance. Another recent performance improvement was made in Tinker-HP that allows for outer timesteps of as long as 10fs, using BAOAB integration, drastically improving the overall simulation speed[207]. Porting this approach over to Tinker-OpenMM should be possible and enable dramatic performance improvements.

The final, most preliminary (but arguably most important) improvement that needs to be made is better sampling approaches. Most current approaches use dynamic approaches based upon the Maxwell kinetic energy distribution. This kinetic energy distribution is necessary in order to capture real-world kinetics. However, this distribution is slower than desired for the calculations of thermodynamic properties such as binding free energies. Standard dynamics base approached also have a tendency to get stuck in alternative minima, as the kinetic energy at room temperature kinetic motion is insufficient to overcome the necessary energy barriers. Improvements to sampling, such as metadynamics approaches[88] or OSRW[91], enable improvements to these aspects by biasing the system's energy surface to enable better sampling. This allows for more rapid and accurate convergence in property calculations (such as free energy calculation).

Over the past several years, GPU computing using Tinker-OpenMM has undergone a transformative change into the main accessible use case for large scale AMOEBA simulations such as protein-ligand systems. The previous advancement in the testing and application of the AMOEBA forcefield has been limited by its poor computational efficiency. GPU computing helps relieve many of the issues associated with this inefficiency. This should allow for improvements in the applications available for the AMOEBA forcefield, as well as incentivize the further development of improved methods.

APPENDIX

Code Details

FREE ENERGY PERTURBATION IMPLEMENTATION DETAILS

On the CPU ommstuff.cpp side, the free energy perturbation changes were minor. For the VdW force, the OpenMM_AmoebaVdwForce_addParticle() routine was modified to give each particle a lambda value(1 if non-ligand, otherwise, the gobal vdw-lambda value. The modifications to the electrostatics on the interface side operated similarly, though no changes needed to be made to the interface. The scaling of multipole factors was already handled by the Tinker CPU reading of variables; no modifications were necessary.

On the GPU side, no changes needed to be made to the electrostatic forces. The vdW changes are in the routine that passes parameters to the GPU (plugins/amoeba/platforms/cuda/src/AmoebaCudaKernels.cpp) in order to pass the lambda array to GPU, and in the GPU kernel

plugins/amoeba/platforms/cuda/src/kernels/Amoebavdwforce2.cu). The kernel changes consisted of the inclusion of a combined lambda variable for each interaction. If two lambdas are non-identical (in the case of a ligand-environment interaction), the lower lambda value (that of the ligand) is used. Otherwise (in ligand-ligand or environment-environment interactions), this variable is set to 1.

RELATIVE FREE ENERGY IMPLEMENTATION

This code is currently in the TinkerRelative and TinkerOMMRelative github branches. The electrostatic lambda is read in from two variables, elamdba1 and elambda2. Each of these keywords is accepted and read in by this version of Tinker, as are the ligand1 and ligand2 keywords that describe ligand bounds. The lambda 1 and 2 are handled by mutate.f, like in the normal build. vdW lambda is handed in a similar manner on the CPU, with each particle given a value of vdwlambda. If in ligand 1, 1-vdwambda if in ligand 2, or 1 otherwise. Therefore, 1 is compete for ligand 1 vdW, and 0 is compete ligand 2 vdW.

On the GPU side, no changes were needed for electrostatic code. As described in the manuscript, this was a decision made in order to make coding feasible. The AmoebaCudaKernels works in a similar manner as normal vdW code, though launched with the amoebaVdwforce2reative kernel. Each atom is put into one of 3 groups. 0 if the environment, 1 if ligand1, and 2 if ligand 2. Each interaction is gated by the statement if abs((2.0-(numGroups1*numGroups2)))>0.1. If atom 1 or 2 is environment or the same ligand, this expression is 2>0.1, or either 1>0.1 or 2>0.1, which always results in execution. However, if an interaction is between a ligand 1 atom and a ligand 2 atom, this expression results in a number close to (though not exactly, due to floating-point math) 0, causing the entire force calculation method to be ignored. This prevents the

interaction of ligand 1 and ligand 2 atoms via the vdW force, which would result in overlap and simulation catastrophe.

VIRIAL CODING DETAILS

The virial code is contained within the VIRIAL branch of github. The actual virial implementation relies on the addition of values to two variables, SlowViral and FastVirial. This separation was done in order to enable multi timestep integrators, and are created in platforms/cuda/src/CudaContext.cpp. This unique location in GPU memory does not move throughout the simulation, allowing for different forces to add to each in a pace in global GPU memory.

One aspect that was made very carefully was ensuring that race condition in the adding to global virial arrays was avoided. For the bonded forces, each bonded force adds to a local Vxx, etc. value. Writes to this local variable do not coincide (this bonded code is not well parallelized). Vxx is then atomically added to the fast virial. The nonbonded forces handle this problem differently, as race conditions are much more likely. The virial additions are thus added directly to central memory. This may lead to some bottlenecking, but this simple atomic addition is a negligible cost.

The virial calculation for most of the forces was relatively trivia porting of code from the CPU. The only real change was in the Ewald virial calculation. It was determined that since the FFT grid needed to be completey different from that used in force and energy computation, the intermediates from this code could not be used. However, the basic code structure from force Ewald was duplicated and modified for virial calculation.

The implemented RESPA code is based upon the implementation in tinker CPU. The CustomIntegrator implementation of the r-RESPA was used in order to ensure that

maximum compatibility and utilization of well-trusted code. The critical change to this code is the addition of a new command (addScaleBox) that performs virial-based coordinate and box size scaling using a modified version of the scaling routine used by the Monte Carlo Barostat. This command was placed in the interface at the end of integration to enable scaling at a time equivalent to that used by Tinker CPU code.

REFERENCES

1. Macalino, S. J. Y.; Gosu, V.; Hong, S.; Choi, S., Role of computer-aided drug design in modern drug discovery. *Archives of pharmacal research* **2015,** *38* (9), 1686-1701.
2. Gaieb, Z.; Liu, S.; Gathiaka, S.; Chiu, M.; Yang, H.; Shao, C.; Feher, V. A.; Walters, W. P.; Kuhn, B.; Rudolph, M. G., D3R Grand Challenge 2: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. *Journal of computer-aided molecular design* **2018,** *32* (1), 1-20.
3. Kitchen, D. B.; Decornez, H.; Furr, J. R.; Bajorath, J., Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery* **2004,** *3* (11), 935.
4. Ferreira, L.; dos Santos, R.; Oliva, G.; Andricopulo, A., Molecular docking and structure-based drug design strategies. *Molecules* **2015,** *20* (7), 13384-13421.
5. Wang, R.; Lu, Y.; Wang, S., Comparative evaluation of 11 scoring functions for molecular docking. *Journal of medicinal chemistry* **2003,** *46* (12), 2287-2303.
6. Friesner, R. A.; Banks, J. L.; Murphy, R. B.; Halgren, T. A.; Klicic, J. J.; Mainz, D. T.; Repasky, M. P.; Knoll, E. H.; Shelley, M.; Perry, J. K., Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *Journal of medicinal chemistry* **2004,** *47* (7), 1739-1749.
7. Verdonk, M. L.; Cole, J. C.; Hartshorn, M. J.; Murray, C. W.; Taylor, R. D., Improved protein–ligand docking using GOLD. *Proteins: Structure, Function, and Bioinformatics* **2003,** *52* (4), 609-623.
8. Verdonk, M. L.; Chessari, G.; Cole, J. C.; Hartshorn, M. J.; Murray, C. W.; Nissink, J. W. M.; Taylor, R. D.; Taylor, R., Modeling water molecules in protein− ligand docking using GOLD. *Journal of medicinal chemistry* **2005,** *48* (20), 6504-6515.
9. Jacq, N.; Salzemann, J.; Legre, Y.; Reichstadt, M.; Jacq, F.; Zimmermann, M.; Maass, A.; Sridhar, M.; Vinod-Kusam, K.; Schwichtenberg, H., Demonstration of in silico docking at a large scale on grid infrastructure. *Studies in health technology and informatics* **2006,** *120*, 155.
10. Vangone, A.; Schaarschmidt, J.; Koukos, P.; Geng, C.; Citro, N.; Trellet, M. E.; Xue, L. C.; Bonvin, A. M., Large-scale prediction of binding affinity in protein–small ligand complexes: the PRODIGY-LIG web server. *Bioinformatics* **2018,** *35* (9), 1585-1587.
11. Li, Y.; Han, L.; Liu, Z.; Wang, R., Comparative assessment of scoring functions on an updated benchmark: 2. Evaluation methods and general results. *Journal of chemical information and modeling* **2014,** *54* (6), 1717-1736.
12. Leach, A. R., Ligand docking to proteins with discrete side-chain flexibility. *Journal of molecular biology* **1994,** *235* (1), 345-356.
13. Nabuurs, S. B.; Wagener, M.; De Vlieg, J., A flexible approach to induced fit docking. *Journal of medicinal chemistry* **2007,** *50* (26), 6507-6518.
14. Sherman, W.; Day, T.; Jacobson, M. P.; Friesner, R. A.; Farid, R., Novel procedure for modeling ligand/receptor induced fit effects. *Journal of medicinal chemistry* **2006,** *49* (2), 534-553.
15. Lavecchia, A., Machine-learning approaches in drug discovery: methods and applications. *Drug discovery today* **2015,** *20* (3), 318-331.
16. Botu, V.; Batra, R.; Chapman, J.; Ramprasad, R., Machine learning force fields: Construction, validation, and outlook. *The Journal of Physical Chemistry C* **2016,** *121* (1), 511-522.

17. Weininger, D., SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* **1988,** *28* (1), 31-36.
18. Maltarollo, V. G.; Gertrudes, J. C.; Oliveira, P. R.; Honorio, K. M., Applying machine learning techniques for ADME-Tox prediction: a review. *Expert opinion on drug metabolism & toxicology* **2015,** *11* (2), 259-271.
19. Sedykh, A.; Fourches, D.; Duan, J.; Hucke, O.; Garneau, M.; Zhu, H.; Bonneau, P.; Tropsha, A., Human intestinal transporter database: QSAR modeling and virtual profiling of drug uptake, efflux and interactions. *Pharmaceutical research* **2013,** *30* (4), 996-1007.
20. Gombar, V. K.; Hall, S. D., Quantitative structure–activity relationship models of clinical pharmacokinetics: clearance and volume of distribution. *Journal of chemical information and modeling* **2013,** *53* (4), 948-957.
21. Cheng, F.; Zhao, Z., Machine learning-based prediction of drug–drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties. *Journal of the American Medical Informatics Association* **2014,** *21* (e2), e278-e286.
22. Ballester, P. J.; Mitchell, J. B., A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics* **2010,** *26* (9), 1169-1175.
23. Klon, A. E.; Glick, M.; Davies, J. W., Application of machine learning to improve the results of high-throughput docking against the HIV-1 protease. *Journal of chemical information and computer sciences* **2004,** *44* (6), 2216-2224.
24. Medina Marrero, R.; Marrero-Ponce, Y.; Barigye, S.; Echeverria Diaz, Y.; Acevedo-Barrios, R.; Casanola-Martin, G.; Garcia Bernal, M.; Torrens, F.; Perez-Gimenez, F., QuBiLs-MAS method in early drug discovery and rational drug identification of antifungal agents. *SAR and QSAR in Environmental Research* **2015,** *26* (11), 943-958.
25. Wang, Y.; Guo, Y.; Kuang, Q.; Pu, X.; Ji, Y.; Zhang, Z.; Li, M., A comparative study of family-specific protein–ligand complex affinity prediction based on random forest approach. *Journal of computer-aided molecular design* **2015,** *29* (4), 349-360.
26. Sakurai, J. J.; Commins, E. D., Modern quantum mechanics, revised edition. AAPT: 1995.
27. Slater, J. C., A simplification of the Hartree-Fock method. *Physical review* **1951,** *81* (3), 385.
28. Møller, C.; Plesset, M. S., Note on an approximation treatment for many-electron systems. *Physical review* **1934,** *46* (7), 618.
29. Knizia, G.; Adler, T. B.; Werner, H.-J., Simplified CCSD (T)-F12 methods: Theory and benchmarks. *The Journal of Chemical Physics* **2009,** *130* (5), 054104.
30. Ratcliff, L. E.; Mohr, S.; Huhs, G.; Deutsch, T.; Masella, M.; Genovese, L., Challenges in large scale quantum mechanical calculations. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2017,** *7* (1), e1290.

31. Raha, K.; Peters, M. B.; Wang, B.; Yu, N.; Wollacott, A. M.; Westerhoff, L. M.; Merz Jr, K. M., The role of quantum mechanics in structure-based drug design. *Drug discovery today* **2007,** *12* (17-18), 725-731.
32. Kaminski, G. A.; Stern, H. A.; Berne, B. J.; Friesner, R. A., Development of an accurate and robust polarizable molecular mechanics force field from ab initio quantum chemistry. *The Journal of Physical Chemistry A* **2004,** *108* (4), 621-627.
33. Parker, T. M.; Burns, L. A.; Parrish, R. M.; Ryno, A. G.; Sherrill, C. D., Levels of symmetry adapted perturbation theory (SAPT). I. Efficiency and performance for interaction energies. *The Journal of chemical physics* **2014,** *140* (9), 094106.
34. Senn, H. M.; Thiel, W., QM/MM methods for biomolecular systems. *Angewandte Chemie International Edition* **2009,** *48* (7), 1198-1229.
35. Olsson, M. A.; Ryde, U., Comparison of QM/MM methods to obtain ligand-binding free energies. *Journal of chemical theory and computation* **2017,** *13* (5), 2245-2253.
36. Grubmüller, H.; Heller, H.; Windemuth, A.; Schulten, K., Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular Simulation* **1991,** *6* (1-3), 121-142.
37. Humphreys, D. D.; Friesner, R. A.; Berne, B. J., A multiple-time-step molecular dynamics algorithm for macromolecules. *The Journal of Physical Chemistry* **1994,** *98* (27), 6885-6892.
38. Hopkins, C. W.; Le Grand, S.; Walker, R. C.; Roitberg, A. E., Long-time-step molecular dynamics through hydrogen mass repartitioning. *Journal of chemical theory and computation* **2015,** *11* (4), 1864-1874.
39. Berendsen, H. J.; Postma, J. v.; van Gunsteren, W. F.; DiNola, A.; Haak, J., Molecular dynamics with coupling to an external bath. *The Journal of chemical physics* **1984,** *81* (8), 3684-3690.
40. Hoover, W. G., Canonical dynamics: Equilibrium phase-space distributions. *Phys Rev A Gen Phys* **1985,** *31* (3), 1695-1697.
41. Feller, S. E.; Zhang, Y.; Pastor, R. W.; Brooks, B. R., Constant pressure molecular dynamics simulation: the Langevin piston method. *The Journal of chemical physics* **1995,** *103* (11), 4613-4621.
42. Bussi, G.; Donadio, D.; Parrinello, M., Canonical sampling through velocity rescaling. *The Journal of chemical physics* **2007,** *126* (1), 014101.
43. Weinan, E.; Li, D., The Andersen thermostat in molecular dynamics. *Communications on Pure and Applied Mathematics* **2008,** *61* (1), 96-136.
44. Roux, B., The calculation of the potential of mean force using computer simulations. *Computer physics communications* **1995,** *91* (1-3), 275-282.
45. Allen, T. W.; Andersen, O. S.; Roux, B., Molecular dynamics—potential of mean force calculations as a tool for understanding ion permeation and selectivity in narrow channels. *Biophysical chemistry* **2006,** *124* (3), 251-267.
46. Tiwary, P.; Limongelli, V.; Salvalaglio, M.; Parrinello, M., Kinetics of protein–ligand unbinding: Predicting pathways, rates, and rate-limiting steps. *Proceedings of the National Academy of Sciences* **2015,** *112* (5), E386-E391.
47. Chodera, J. D.; Mobley, D. L.; Shirts, M. R.; Dixon, R. W.; Branson, K.; Pande, V. S., Alchemical free energy methods for drug discovery: progress and challenges. *Current opinion in structural biology* **2011,** *21* (2), 150-160.

48. Bennett, C. H., Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics* **1976,** *22* (2), 245-268.
49. Straatsma, T.; Berendsen, H., Free energy of ionic hydration: Analysis of a thermodynamic integration technique to evaluate free energy differences by molecular dynamics simulations. *The Journal of chemical physics* **1988,** *89* (9), 5876-5886.
50. Force, T.; Krause, D. S.; Van Etten, R. A., Molecular mechanisms of cardiotoxicity of tyrosine kinase inhibition. *Nature Reviews Cancer* **2007,** *7* (5), 332.
51. Pearlman, D. A.; Case, D. A.; Caldwell, J. W.; Ross, W. S.; Cheatham III, T. E.; DeBolt, S.; Ferguson, D.; Seibel, G.; Kollman, P., AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications* **1995,** *91* (1-3), 1-41.
52. Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S. a.; Karplus, M., CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *Journal of computational chemistry* **1983,** *4* (2), 187-217.
53. Toukmaji, A. Y.; Board Jr, J. A., Ewald summation techniques in perspective: a survey. *Computer physics communications* **1996,** *95* (2-3), 73-92.
54. Williams, D. E., Representation of the molecular electrostatic potential by atomic multipole and bond dipole models. *Journal of computational chemistry* **1988,** *9* (7), 745-763.
55. Clark, T.; Hennemann, M.; Murray, J. S.; Politzer, P., Halogen bonding: the σ-hole. *Journal of molecular modeling* **2007,** *13* (2), 291-296.
56. Scholfield, M. R.; Zanden, C. M. V.; Carter, M.; Ho, P. S., Halogen bonding (X-bonding): A biological perspective. *Protein Science* **2013,** *22* (2), 139-152.
57. Mu, X.; Wang, Q.; Wang, L.-P.; Fried, S. D.; Piquemal, J.-P.; Dalby, K. N.; Ren, P., Modeling organochlorine compounds and the σ-hole effect using a polarizable multipole force field. *The Journal of Physical Chemistry B* **2014,** *118* (24), 6456-6465.
58. Rick, S. W.; Stuart, S. J.; Berne, B. J., Dynamical fluctuating charge force fields: Application to liquid water. *The Journal of chemical physics* **1994,** *101* (7), 6141-6156.
59. Patel, S.; Mackerell Jr, A. D.; Brooks III, C. L., CHARMM fluctuating charge force field for proteins: II protein/solvent properties from molecular dynamics simulations using a nonadditive electrostatic model. *Journal of computational chemistry* **2004,** *25* (12), 1504-1514.
60. Anisimov, V. M.; Lamoureux, G.; Vorobyov, I. V.; Huang, N.; Roux, B.; MacKerell, A. D., Determination of electrostatic parameters for a polarizable force field based on the classical Drude oscillator. *Journal of Chemical Theory and Computation* **2005,** *1* (1), 153-168.
61. Lopes, P. E.; Huang, J.; Shim, J.; Luo, Y.; Li, H.; Roux, B.; MacKerell Jr, A. D., Polarizable force field for peptides and proteins based on the classical drude oscillator. *Journal of chemical theory and computation* **2013,** *9* (12), 5430-5449.

62. Owens, J. D.; Houston, M.; Luebke, D.; Green, S.; Stone, J. E.; Phillips, J. C., GPU computing. **2008**.
63. Plimpton, S.; Pollock, R.; Stevens, M. In *Particle-Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations*, PPSC, Citeseer: 1997.
64. Harger, M.; Li, D.; Wang, Z.; Dalby, K.; Lagardère, L.; Piquemal, J. P.; Ponder, J.; Ren, P., Tinker-OpenMM: Absolute and relative alchemical free energies using AMOEBA on GPUs. *Journal of computational chemistry* **2017,** *38* (23), 2047-2055.
65. Eastman, P.; Pande, V., OpenMM: a hardware-independent framework for molecular simulations. *Computing in Science & Engineering* **2010,** *12* (4), 34-39.
66. Eastman, P.; Friedrichs, M. S.; Chodera, J. D.; Radmer, R. J.; Bruns, C. M.; Ku, J. P.; Beauchamp, K. A.; Lane, T. J.; Wang, L.-P.; Shukla, D., OpenMM 4: a reusable, extensible, hardware independent library for high performance molecular simulation. *Journal of chemical theory and computation* **2012,** *9* (1), 461-469.
67. Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D., OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology* **2017,** *13* (7), e1005659.
68. Munshi, A. In *The opencl specification*, 2009 IEEE Hot Chips 21 Symposium (HCS), IEEE: 2009; pp 1-314.
69. Nvidia, C., Programming guide. 2010.
70. Sanz, E.; Vega, C., Solubility of KF and NaCl in water by molecular simulation. *The Journal of chemical physics* **2007,** *126* (1), 014507.
71. Seeliger, D.; De Groot, B. L., Protein thermostability calculations using alchemical free energy simulations. *Biophysical journal* **2010,** *98* (10), 2309-2316.
72. Aqvist, J., Ion-water interaction potentials derived from free energy perturbation simulations. *The Journal of Physical Chemistry* **1990,** *94* (21), 8021-8024.
73. Garrido, N. M.; Queimada, A. J.; Jorge, M.; Macedo, E. A.; Economou, I. G., 1-octanol/water partition coefficients of n-alkanes from molecular simulations of absolute solvation free energies. *Journal of Chemical Theory and Computation* **2009,** *5* (9), 2436-2446.
74. Jayaraman, S.; Maginn, E. J., Computing the melting point and thermodynamic stability of the orthorhombic and monoclinic crystalline polymorphs of the ionic liquid 1-n-butyl-3-methylimidazolium chloride. *The Journal of chemical physics* **2007,** *127* (21), 214504.
75. Warren, G. L.; Andrews, C. W.; Capelli, A.-M.; Clarke, B.; LaLonde, J.; Lambert, M. H.; Lindvall, M.; Nevins, N.; Semus, S. F.; Senger, S., A critical assessment of docking programs and scoring functions. *Journal of medicinal chemistry* **2006,** *49* (20), 5912-5931.
76. Fischer, M.; Coleman, R. G.; Fraser, J. S.; Shoichet, B. K., Incorporation of protein flexibility and conformational energy penalties in docking screens to improve ligand discovery. *Nature chemistry* **2014,** *6* (7), 575.
77. Enyedy, I. J.; Egan, W. J., Can we use docking and scoring for hit-to-lead optimization? *Journal of computer-aided molecular design* **2008,** *22* (3-4), 161-168.
78. Jorgensen, W. L., The many roles of computation in drug discovery. *Science* **2004,** *303* (5665), 1813-1818.

79. Mobley, D. L.; Heinzelmann, G.; Henriksen, N. M.; Gilson, M. K., Predicting binding free energies: Frontiers and benchmarks (a perpetual review). **2017**.
80. Gilson, M. K.; Zhou, H.-X., Calculation of protein-ligand binding affinities. *Annu. Rev. Biophys. Biomol. Struct.* **2007,** *36*, 21-42.
81. Jorgensen, W. L.; Buckner, J. K.; Boudon, S.; Tirado-Rives, J., Efficient computation of absolute free energies of binding by computer simulations. Application to the methane dimer in water. *The Journal of chemical physics* **1988,** *89* (6), 3742-3746.
82. Gilson, M. K.; Given, J. A.; Bush, B. L.; McCammon, J. A., The statistical-thermodynamic basis for computation of binding affinities: a critical review. *Biophysical journal* **1997,** *72* (3), 1047-1069.
83. Izrailev, S.; Stepaniants, S.; Isralewitz, B.; Kosztin, D.; Lu, H.; Molnar, F.; Wriggers, W.; Schulten, K., Computational Molecular Dynamics: Challenges, Methods, Ideas, Lecture Notes in Computational Science and Engineering. **1998**.
84. Gullingsrud, J. R.; Braun, R.; Schulten, K., Reconstructing potentials of mean force through time series analysis of steered molecular dynamics simulations. *Journal of Computational Physics* **1999,** *151* (1), 190-211.
85. Zhang, D.; Gullingsrud, J.; McCammon, J. A., Potentials of mean force for acetylcholine unbinding from the alpha7 nicotinic acetylcholine receptor ligand-binding domain. *Journal of the American Chemical Society* **2006,** *128* (9), 3019-3026.
86. Woo, H.-J.; Roux, B., Calculation of absolute protein–ligand binding free energy from computer simulations. *Proceedings of the National Academy of Sciences* **2005,** *102* (19), 6825-6830.
87. Lau, A. Y.; Roux, B., The hidden energetics of ligand binding and activation in a glutamate receptor. *Nature structural & molecular biology* **2011,** *18* (3), 283.
88. Barducci, A.; Bussi, G.; Parrinello, M., Well-tempered metadynamics: a smoothly converging and tunable free-energy method. *Physical review letters* **2008,** *100* (2), 020603.
89. Laio, A.; Gervasio, F. L., Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. *Reports on Progress in Physics* **2008,** *71* (12), 126601.
90. Bussi, G.; Laio, A.; Parrinello, M., Equilibrium free energies from nonequilibrium metadynamics. *Physical review letters* **2006,** *96* (9), 090601.
91. Zheng, L.; Chen, M.; Yang, W., Random walk in orthogonal space to achieve efficient free-energy simulation of complex systems. *Proceedings of the National Academy of Sciences* **2008,** *105* (51), 20227-20232.
92. Zheng, L.; Chen, M.; Yang, W., Simultaneous escaping of explicit and hidden free energy barriers: Application of the orthogonal space random walk strategy in generalized ensemble based conformational sampling. *The Journal of chemical physics* **2009,** *130* (23), 06B618.
93. Shirts, M. R.; Pande, V. S., Comparison of efficiency and bias of free energies computed by exponential averaging, the Bennett acceptance ratio, and thermodynamic integration. *The Journal of chemical physics* **2005,** *122* (14), 144107.

94. Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I., CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *Journal of computational chemistry* **2010,** *31* (4), 671-690.
95. Klauda, J. B.; Venable, R. M.; Freites, J. A.; O'Connor, J. W.; Tobias, D. J.; Mondragon-Ramirez, C.; Vorobyov, I.; MacKerell Jr, A. D.; Pastor, R. W., Update of the CHARMM all-atom additive force field for lipids: validation on six lipid types. *The journal of physical chemistry B* **2010,** *114* (23), 7830-7843.
96. MacKerell Jr, A. D.; Banavali, N.; Foloppe, N., Development and current status of the CHARMM force field for nucleic acids. *Biopolymers: Original Research on Biomolecules* **2000,** *56* (4), 257-265.
97. Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A., Development and testing of a general amber force field. *Journal of computational chemistry* **2004,** *25* (9), 1157-1174.
98. Pérez, A.; Marchán, I.; Svozil, D.; Sponer, J.; Cheatham III, T. E.; Laughton, C. A.; Orozco, M., Refinement of the AMBER force field for nucleic acids: improving the description of α/γ conformers. *Biophysical journal* **2007,** *92* (11), 3817-3829.
99. Meagher, K. L.; Redman, L. T.; Carlson, H. A., Development of polyphosphate parameters for use with the AMBER force field. *Journal of computational chemistry* **2003,** *24* (9), 1016-1025.
100. Ren, P.; Ponder, J. W., Polarizable atomic multipole water model for molecular mechanics simulation. *The Journal of Physical Chemistry B* **2003,** *107* (24), 5933-5947.
101. Wu, J. C.; Chattree, G.; Ren, P., Automation of AMOEBA polarizable force field parameterization for small molecules. *Theoretical chemistry accounts* **2012,** *131* (3), 1138.
102. Shi, Y.; Xia, Z.; Zhang, J.; Best, R.; Wu, C.; Ponder, J. W.; Ren, P., Polarizable atomic multipole-based AMOEBA force field for proteins. *Journal of chemical theory and computation* **2013,** *9* (9), 4046-4063.
103. Maple, J. R.; Cao, Y.; Damm, W.; Halgren, T. A.; Kaminski, G. A.; Zhang, L. Y.; Friesner, R. A., A polarizable force field and continuum solvation methodology for modeling of protein− ligand interactions. *Journal of Chemical Theory and Computation* **2005,** *1* (4), 694-715.
104. Stern, H. A.; Kaminski, G. A.; Banks, J. L.; Zhou, R.; Berne, B.; Friesner, R. A., Fluctuating charge, polarizable dipole, and combined models: parameterization from ab initio quantum chemistry. *The Journal of Physical Chemistry B* **1999,** *103* (22), 4730-4737.
105. Patel, S.; Brooks III, C. L., CHARMM fluctuating charge force field for proteins: I parameterization and application to bulk organic liquid simulations. *Journal of computational chemistry* **2004,** *25* (1), 1-16.
106. Baker, C. M.; Anisimov, V. M.; MacKerell Jr, A. D., Development of CHARMM polarizable force field for nucleic acid bases based on the classical Drude oscillator model. *The journal of physical chemistry B* **2010,** *115* (3), 580-596.
107. Lopes, P. E.; Lamoureux, G.; Roux, B.; MacKerell, A. D., Polarizable empirical force field for aromatic compounds based on the classical drude oscillator. *The Journal of*

Physical Chemistry B **2007,** *111* (11), 2873-2885.
108. Baker, C. M.; Lopes, P. E.; Zhu, X.; Roux, B.; MacKerell Jr, A. D., Accurate calculation of hydration free energies using pair-specific Lennard-Jones parameters in the CHARMM Drude polarizable force field. *Journal of chemical theory and computation* **2010,** *6* (4), 1181-1198.
109. Ren, P.; Wu, C.; Ponder, J. W., Polarizable atomic multipole-based molecular mechanics for organic molecules. *Journal of chemical theory and computation* **2011,** *7* (10), 3143-3161.
110. Shi, Y.; Wu, C.; Ponder, J. W.; Ren, P., Multipole electrostatics in hydration free energy calculations. *Journal of computational chemistry* **2011,** *32* (5), 967-977.
111. Abella, J. R.; Cheng, S. Y.; Wang, Q.; Yang, W.; Ren, P., Hydration free energy from orthogonal space random walk and polarizable force field. *Journal of chemical theory and computation* **2014,** *10* (7), 2792-2801.
112. Schnieders, M. J.; Baltrusaitis, J.; Shi, Y.; Chattree, G.; Zheng, L.; Yang, W.; Ren, P., The structure, thermodynamics, and solubility of organic crystals from simulation with a polarizable force field. *Journal of chemical theory and computation* **2012,** *8* (5), 1721-1736.
113. Jiao, D.; King, C.; Grossfield, A.; Darden, T. A.; Ren, P., Simulation of Ca2+ and Mg2+ solvation using polarizable atomic multipole potential. *The Journal of Physical Chemistry B* **2006,** *110* (37), 18553-18559.
114. Wu, J. C.; Piquemal, J.-P.; Chaudret, R.; Reinhardt, P.; Ren, P., Polarizable molecular dynamics simulation of Zn (II) in water using the AMOEBA force field. *Journal of chemical theory and computation* **2010,** *6* (7), 2059-2070.
115. Grossfield, A.; Ren, P.; Ponder, J. W., Ion solvation thermodynamics from simulation with a polarizable force field. *Journal of the American Chemical Society* **2003,** *125* (50), 15671-15682.
116. Bell, D. R.; Qi, R.; Jing, Z.; Xiang, J. Y.; Mejias, C.; Schnieders, M. J.; Ponder, J. W.; Ren, P., Calculating binding free energies of host–guest systems using the AMOEBA polarizable force field. *Physical Chemistry Chemical Physics* **2016,** *18* (44), 30261-30269.
117. Zhang, J.; Shi, Y.; Ren, P., Polarizable force fields for scoring protein–ligand interactions. *Protein-Ligand Interactions* **2012**, 99-120.
118. Shi, Y.; Zhu, C. Z.; Martin, S. F.; Ren, P., Probing the effect of conformational constraint on phosphorylated ligand binding to an SH2 domain using polarizable force field simulations. *The Journal of Physical Chemistry B* **2012,** *116* (5), 1716-1727.
119. Jiao, D.; Zhang, J.; Duke, R. E.; Li, G.; Schnieders, M. J.; Ren, P., Trypsin-ligand binding free energies from explicit and implicit solvent simulations with polarizable potential. *Journal of computational chemistry* **2009,** *30* (11), 1701-1711.
120. Jiao, D.; Golubkov, P. A.; Darden, T. A.; Ren, P., Calculation of protein–ligand binding free energy by using a polarizable potential. *Proceedings of the National Academy of Sciences* **2008,** *105* (17), 6290-6295.
121. Zhang, J.; Yang, W.; Piquemal, J.-P.; Ren, P., Modeling structural coordination and ligand binding in zinc proteins with a polarizable potential. *Journal of chemical*

theory and computation **2012,** *8* (4), 1314-1324.
122. Dagum, L., R, Menon, OpenMP: An Industry-Standard API for Shared-Memory Programming, Computational Science & Engineering, Vol. 5, No. 1. *IEEE January/March* **1998**.
123. Narth, C.; Lagardère, L.; Polack, E.; Gresh, N.; Wang, Q.; Bell, D. R.; Rackers, J. A.; Ponder, J. W.; Ren, P. Y.; Piquemal, J. P., Scalable improvement of SPME multipolar electrostatics in anisotropic polarizable molecular mechanics using a general short-range penetration correction up to quadrupoles. *Journal of computational chemistry* **2016,** *37* (5), 494-506.
124. Lipparini, F.; Lagardère, L.; Stamm, B.; Cancès, E.; Schnieders, M.; Ren, P.; Maday, Y.; Piquemal, J.-P., Scalable evaluation of polarization energy and associated forces in polarizable molecular dynamics: I. toward massively parallel direct space computations. *Journal of Chemical Theory and Computation* **2014,** *10* (4), 1638-1651.
125. Levitt, M., Protein folding by restrained energy minimization and molecular dynamics. *Journal of molecular biology* **1983,** *170* (3), 723-764.
126. Hornak, V.; Simmerling, C., Development of softcore potential functions for overcoming steric barriers in molecular dynamics simulations. *Journal of Molecular Graphics and Modelling* **2004,** *22* (5), 405-413.
127. Salomon-Ferrer, R.; Case, D.; Walker, R., An overview of the Amber biomolecular simulation package. 2012, WIREs Comput. *Mol. Sci*.
128. Phillips, J. C.; Stone, J. E.; Schulten, K. In *Adapting a message-driven parallel application to GPU-accelerated clusters*, Proceedings of the 2008 ACM/IEEE conference on Supercomputing, IEEE Press: 2008; p 8.
129. Jorgensen, W. L.; Nguyen, T. B., Monte Carlo simulations of the hydration of substituted benzenes with OPLS potential functions. *Journal of Computational Chemistry* **1993,** *14* (2), 195-205.
130. Ota, N.; Stroupe, C.; Ferreira-da-Silva, J.; Shah, S. A.; Mares-Guia, M.; Brunger, A. T., Non-Boltzmann thermodynamic integration (NBTI) for macromolecular systems: Relative free energy of binding of trypsin to benzamidine and benzylamine. *Proteins: Structure, Function, and Bioinformatics* **1999,** *37* (4), 641-653.
131. Miyamoto, S.; Kollman, P. A., Absolute and relative binding free energy calculations of the interaction of biotin and its analogs with streptavidin using molecular dynamics/free energy perturbation approaches. *Proteins: Structure, Function, and Bioinformatics* **1993,** *16* (3), 226-245.
132. Reddy, M. R.; Erion, M. D., Calculation of relative binding free energy differences for fructose 1, 6-bisphosphatase inhibitors using the thermodynamic cycle perturbation approach. *Journal of the American Chemical Society* **2001,** *123* (26), 6246-6252.
133. Reddy, M. R.; Viswanadhan, V. N.; Weinstein, J. N., Relative differences in the binding free energies of human immunodeficiency virus 1 protease inhibitors: a thermodynamic cycle-perturbation approach. *Proceedings of the National Academy of Sciences* **1991,** *88* (22), 10287-10291.

134. Fleischman, S. H.; Brooks III, C. L., Thermodynamics of aqueous solvation: Solution properties of alcohols and alkanes. *The Journal of chemical physics* **1987,** *87* (5), 3029-3037.
135. Ponder, J.; Richards, F., TINKER molecular modeling package. *J. Comput. Chem* **1987,** *8*, 1016-1024.
136. Zhang, C.; Bell, D.; Harger, M.; Ren, P., Polarizable Multipole-Based Force Field for Aromatic Molecules and Nucleobases. *Journal of chemical theory and computation* **2017,** *13* (2), 666-678.
137. Muddana, H. S.; Fenley, A. T.; Mobley, D. L.; Gilson, M. K., The SAMPL4 host–guest blind prediction challenge: an overview. *Journal of computer-aided molecular design* **2014,** *28* (4), 305-317.
138. Roux, B.; Nina, M.; Pomes, R.; Smith, J. C., Thermodynamic stability of water molecules in the bacteriorhodopsin proton channel: a molecular dynamics free energy perturbation study. *Biophysical journal* **1996,** *71* (2), 670-681.
139. Harger, M.; Lee, J. H.; Walker, B.; Taliaferro, J. M.; Edupuganti, R.; Dalby, K. N.; Ren, P., Computational insights into the binding of IN17 inhibitors to MELK. *J Mol Model* **2019,** *25* (6), 151.
140. Cohen, P., Protein kinases—the major drug targets of the twenty-first century? *Nature reviews Drug discovery* **2002,** *1* (4), 309.
141. Vazquez-Martin, A.; Oliveras-Ferraros, C.; Menendez, J. A., The active form of the metabolic sensor AMP-activated protein kinase α (AMPKα) directly binds the mitotic apparatus and travels from centrosomes to the spindle midzone during mitosis and cytokinesis. *Cell cycle* **2009,** *8* (15), 2385-2398.
142. Hut, H. M.; Lemstra, W.; Blaauw, E. H.; Van Cappellen, G. W.; Kampinga, H. H.; Sibon, O. C., Centrosomes split in the presence of impaired DNA integrity during mitosis. *Molecular biology of the cell* **2003,** *14* (5), 1993-2004.
143. Schubert, D., Synergistic interactions between transforming growth factor beta and fibroblast growth factor regulate Schwann cell mitosis. *Journal of neurobiology* **1992,** *23* (2), 143-148.
144. Forbes, S. A.; Bindal, N.; Bamford, S.; Cole, C.; Kok, C. Y.; Beare, D.; Jia, M.; Shepherd, R.; Leung, K.; Menzies, A., COSMIC: mining complete cancer genomes in the Catalogue of Somatic Mutations in Cancer. *Nucleic acids research* **2010,** *39* (suppl_1), D945-D950.
145. Downward, J., Targeting RAS signalling pathways in cancer therapy. *Nature Reviews Cancer* **2003,** *3* (1), 11.
146. Edupuganti, R.; Taliaferro, J. M.; Wang, Q.; Xie, X.; Cho, E. J.; Vidhu, F.; Ren, P.; Anslyn, E. V.; Bartholomeusz, C.; Dalby, K. N., Discovery of a potent inhibitor of MELK that inhibits expression of the anti-apoptotic protein Mcl-1 and TNBC cell growth. *Bioorgan Med Chem* **2017,** *25* (9), 2609-2616.

147. Gray, D.; Jubb, A. M.; Hogue, D.; Dowd, P.; Kljavin, N.; Yi, S.; Bai, W.; Frantz, G.; Zhang, Z.; Koeppen, H.; de Sauvage, F. J.; Davis, D. P., Maternal embryonic leucine zipper kinase/murine protein serine-threonine kinase 38 is a promising therapeutic target for multiple cancers. *Cancer Res* **2005,** *65* (21), 9751-61.
148. Li, Y. F.; Tang, H.; Sun, Z. F.; Bungum, A. O.; Edell, E. S.; Lingle, W. L.; Stoddard, S. M.; Zhang, M. R.; Jen, J.; Yang, P.; Wang, L., Network-based approach identified cell cycle genes as predictor of overall survival in lung adenocarcinoma patients. *Lung Cancer* **2013,** *80* (1), 91-98.
149. Ryu, B.; Kim, D. S.; Deluca, A. M.; Alani, R. M., Comprehensive expression profiling of tumor cell lines identifies molecular signatures of melanoma progression. *PLoS One* **2007,** *2* (7), e594.
150. Ganguly, R.; Hong, C. S.; Smith, L. G.; Kornblum, H. I.; Nakano, I., Maternal embryonic leucine zipper kinase: key kinase for stem cell phenotype in glioma and other cancers. *Mol Cancer Ther* **2014,** *13* (6), 1393-8.
151. Chung, S.; Nakamura, Y., MELK inhibitor, novel molecular targeted therapeutics for human cancer stem cells. *Cell Cycle* **2013,** *12* (11), 1655-6.
152. Chung, S. Y.; Suzuki, H.; Miyamoto, T.; Takamatsu, N.; Tatsuguchi, A.; Ueda, K.; Kijima, K.; Nakamura, Y.; Matsuo, Y., Development of an orally-administrative MELK-targeting inhibitor that suppresses the growth of various types of human cancer. *Oncotarget* **2012,** *3* (12), 1629-1640.
153. Ji, W. B.; Arnst, C.; Tipton, A. R.; Bekier, M. E.; Taylor, W. R.; Yen, T. J.; Liu, S. T., OTSSP167 Abrogates Mitotic Checkpoint through Inhibiting Multiple Mitotic Kinases. *Plos One* **2016,** *11* (4), e0153518.
154. Joshi, K.; Banasavadi-Siddegowda, Y.; Mo, X.; Kim, S. H.; Mao, P.; Kig, C.; Nardini, D.; Sobol, R. W.; Chow, L. M.; Kornblum, H. I.; Waclaw, R.; Beullens, M.; Nakano, I., MELK-dependent FOXM1 phosphorylation is essential for proliferation of glioma stem cells. *Stem Cells* **2013,** *31* (6), 1051-63.
155. Gu, C.; Banasavadi-Siddegowda, Y. K.; Joshi, K.; Nakamura, Y.; Kurt, H.; Gupta, S.; Nakano, I., Tumor-specific activation of the C-JUN/MELK pathway regulates glioma stem cell growth in a p53-dependent manner. *Stem Cells* **2013,** *31* (5), 870-81.
156. Wang, Y.; Begley, M.; Li, Q.; Huang, H. T.; Lako, A.; Eck, M. J.; Gray, N. S.; Mitchison, T. J.; Cantley, L. C.; Zhao, J. J., Mitotic MELK-eIF4B signaling controls protein synthesis and tumor cell survival. *Proc Natl Acad Sci U S A* **2016,** *113* (35), 9810-5.
157. Komatsu, M.; Yoshimaru, T.; Matsuo, T.; Kiyotani, K.; Miyoshi, Y.; Tanahashi, T.; Rokutan, K.; Yamaguchi, R.; Saito, A.; Imoto, S.; Miyano, S.; Nakamura, Y.; Sasa, M.; Shimada, M.; Katagiri, T., Molecular features of triple negative breast cancer cells by genome-wide gene expression profiling analysis. *Int J Oncol* **2013,** *42* (2), 478-506.

158. Al-Ejeh, F.; Simpson, P. T.; Saunus, J. M.; Klein, K.; Kalimutho, M.; Shi, W.; Miranda, M.; Kutasovic, J.; Raghavendra, A.; Madore, J.; Reid, L.; Krause, L.; Chenevix-Trench, G.; Lakhani, S. R.; Khanna, K. K., Meta-analysis of the global gene expression profile of triple-negative breast cancer identifies genes for the prognostication and treatment of aggressive breast cancer. *Oncogenesis* **2014,** *3*, e124.
159. Kuner, R.; Falth, M.; Pressinotti, N. C.; Brase, J. C.; Puig, S. B.; Metzger, J.; Gade, S.; Schafer, G.; Bartsch, G.; Steiner, E.; Klocker, H.; Sultmann, H., The maternal embryonic leucine zipper kinase (MELK) is upregulated in high-grade prostate cancer. *J Mol Med* **2013,** *91* (2), 237-248.
160. Alachkar, H.; Mutonga, M. B.; Metzeler, K. H.; Fulton, N.; Malnassy, G.; Herold, T.; Spiekermann, K.; Bohlander, S. K.; Hiddemann, W.; Matsuo, Y.; Stock, W.; Nakamura, Y., Preclinical efficacy of maternal embryonic leucine-zipper kinase (MELK) inhibition in acute myeloid leukemia. *Oncotarget* **2014,** *5* (23), 12371-82.
161. Taylor, S. S.; McKeon, F., Kinetochore localization of murine Bub1 is required for normal mitotic timing and checkpoint response to spindle damage. *Cell* **1997,** *89* (5), 727-735.
162. Wang, F.; Dai, J.; Daum, J. R.; Niedzialkowska, E.; Banerjee, B.; Stukenberg, P. T.; Gorbsky, G. J.; Higgins, J. M., Histone H3 Thr-3 phosphorylation by Haspin positions Aurora B at centromeres in mitosis. *Science* **2010,** *330* (6001), 231-235.
163. Hirota, T.; Lipp, J. J.; Toh, B.-H.; Peters, J.-M., Histone H3 serine 10 phosphorylation by Aurora B causes HP1 dissociation from heterochromatin. *Nature* **2005,** *438* (7071), 1176.
164. Hilberg, F.; Roth, G. J.; Krssak, M.; Kautschitsch, S.; Sommergruber, W.; Tontsch-Grunt, U.; Garin-Chesa, P.; Bader, G.; Zoephel, A.; Quant, J.; Heckel, A.; Rettig, W. J., BIBF 1120: Triple angiokinase inhibitor with sustained receptor blockade and good antitumor efficacy. *Cancer Research* **2008,** *68* (12), 4774-4782.
165. Mobley, D. L.; Gilson, M. K., Predicting binding free energies: frontiers and benchmarks. *Annual review of biophysics* **2017,** *46*, 531-558.
166. Shi, Y.; Xia, Z.; Zhang, J.; Best, R.; Wu, C.; Ponder, J. W.; Ren, P., The Polarizable Atomic Multipole-based AMOEBA Force Field for Proteins. *J Chem Theory Comput* **2013,** *9* (9), 4046-4063.
167. Frisch, M.; Trucks, G.; Schlegel, H.; Scuseria, G.; Robb, M.; Cheeseman, J.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G., Gaussian 09, revision b. 01, Gaussian. *Inc., Wallingford, CT* **2010,** *6492*.
168. Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R., Development and validation of a genetic algorithm for flexible docking. *Journal of molecular biology* **1997,** *267* (3), 727-748.
169. Cao, L.-S.; Wang, J.; Chen, Y.; Deng, H.; Wang, Z.-X.; Wu, J.-W., Structural basis for the regulation of maternal embryonic leucine zipper kinase. *PloS one* **2013,** *8* (7), e70031.
170. Eswar, N.; Webb, B.; Marti-Renom, M. A.; Madhusudhan, M.; Eramian, D.; Shen, M. y.; Pieper, U.; Sali, A., Comparative protein structure modeling using Modeller.

Current protocols in bioinformatics **2006,** *15* (1), 5.6. 1-5.6. 30.
171. Hanwell, M. D.; Curtis, D. E.; Lonie, D. C.; Vandermeersch, T.; Zurek, E.; Hutchison, G. R., Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *Journal of cheminformatics* **2012,** *4* (1), 17.
172. Cances, E.; Mennucci, B.; Tomasi, J., A new integral equation formalism for the polarizable continuum model: Theoretical background and applications to isotropic and anisotropic dielectrics. *The Journal of chemical physics* **1997,** *107* (8), 3032-3041.
173. Richeldi, L.; Du Bois, R. M.; Raghu, G.; Azuma, A.; Brown, K. K.; Costabel, U.; Cottin, V.; Flaherty, K. R.; Hansell, D. M.; Inoue, Y., Efficacy and safety of nintedanib in idiopathic pulmonary fibrosis. *New England Journal of Medicine* **2014,** *370* (22), 2071-2082.
174. Hilberg, F.; Roth, G. J.; Krssak, M.; Kautschitsch, S.; Sommergruber, W.; Tontsch-Grunt, U.; Garin-Chesa, P.; Bader, G.; Zoephel, A.; Quant, J., BIBF 1120: triple angiokinase inhibitor with sustained receptor blockade and good antitumor efficacy. *Cancer research* **2008,** *68* (12), 4774-4782.
175. Canevari, G.; Re Depaolini, S.; Cucchi, U.; Bertrand, J. A.; Casale, E.; Perrera, C.; Forte, B.; Carpinelli, P.; Felder, E. R., Structural insight into maternal embryonic leucine zipper kinase (MELK) conformation and inhibition toward structure-based drug design. *Biochemistry* **2013,** *52* (37), 6380-6387.
176. Klaeger, S.; Heinzlmeir, S.; Wilhelm, M.; Polzer, H.; Vick, B.; Koenig, P.-A.; Reinecke, M.; Ruprecht, B.; Petzoldt, S.; Meng, C., The target landscape of clinical kinase drugs. *Science* **2017,** *358* (6367), eaan4368.
177. Qi, R.; Jing, Z.; Liu, C.; Piquemal, J.-P.; Dalby, K. N.; Ren, P., Elucidating the Phosphate Binding Mode of PBP: The Critical Effect of Buffer Solution. *The Journal of Physical Chemistry B* **2018**.
178. Manning, G.; Whyte, D. B.; Martinez, R.; Hunter, T.; Sudarsanam, S., The protein kinase complement of the human genome. *Science* **2002,** *298* (5600), 1912-1934.
179. Verheul, H. M.; Pinedo, H. M., Possible molecular mechanisms involved in the toxicity of angiogenesis inhibition. *Nature Reviews Cancer* **2007,** *7* (6), 475.
180. Kerkela, R.; Woulfe, K. C.; Durand, J. B.; Vagnozzi, R.; Kramer, D.; Chu, T. F.; Beahm, C.; Chen, M. H.; Force, T., Sunitinib-induced cardiotoxicity is mediated by off-target inhibition of AMP-activated protein kinase. *Clinical and translational science* **2009,** *2* (1), 15-25.
181. Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P., Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology* **2011,** *29* (11), 1046.
182. Laskowski, R. A.; Swindells, M. B., LigPlot+: multiple ligand–protein interaction diagrams for drug discovery. ACS Publications: 2011.
183. Harger, M.; Ren, P., Virial-based Berendsen barostat on GPUs using AMOEBA in Tinker-OpenMM. *Results in Chemistry* **2019,** *1*, 100004.
184. Ghosh, D. B.; Karki, B. B.; Stixrude, L., First-principles molecular dynamics simulations of MgSiO3 glass: Structure, density, and elasticity at high pressure.

American Mineralogist **2014,** *99* (7), 1304-1314.
185. Dong, Y.; Rismiller, S. C.; Lin, J., Molecular dynamic simulation of layered graphene clusters formation from polyimides under extreme conditions. *Carbon* **2016,** *104*, 47-55.
186. Fomin, Y. D.; Ryzhov, V. N.; Tsiok, E. N.; Brazhkin, V. V.; Trachenko, K., Dynamic transition in supercritical iron. *Sci Rep* **2014,** *4*, 7194.
187. Gubin, S.; Maklashova, I.; Selezenev, A.; Kozlova, S., Molecular-dynamics study melting aluminum at high pressures. *Physics Procedia* **2015,** *72*, 338-341.
188. Caro, J. A.; Wand, A. J., Practical aspects of high-pressure NMR spectroscopy and its applications in protein biophysics and structural biology. *Methods* **2018,** *148*, 67-80.
189. Ichiye, T. In *Enzymes from piezophiles*, Seminars in cell & developmental biology, Elsevier: 2018; pp 138-146.
190. Huang, Q.; Rodgers, J. M.; Hemley, R. J.; Ichiye, T., Extreme biophysics: Enzymes under pressure. *J Comput Chem* **2017,** *38* (15), 1174-1182.
191. Wright, P. C.; Westacott, R. E.; Burja, A. M., Piezotolerance as a metabolic engineering tool for the biosynthesis of natural products. *Biomolecular engineering* **2003,** *20* (4-6), 325-331.
192. Tsai, D., The virial theorem and stress calculation in molecular dynamics. *The Journal of Chemical Physics* **1979,** *70* (3), 1375-1382.
193. Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G., A smooth particle mesh Ewald method. *The Journal of chemical physics* **1995,** *103* (19), 8577-8593.
194. Martyna, G. J.; Klein, M. L.; Tuckerman, M., Nosé–Hoover chains: The canonical ensemble via continuous dynamics. *The Journal of chemical physics* **1992,** *97* (4), 2635-2643.
195. Nosé, S., A unified formulation of the constant temperature molecular dynamics methods. *The Journal of chemical physics* **1984,** *81* (1), 511-519.
196. Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L. P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S., OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput Biol* **2017,** *13* (7), e1005659.
197. Ponder, J. W., TINKER: Software tools for molecular design. Version: 2004.
198. Rackers, J. A.; Wang, Z.; Lu, C.; Laury, M. L.; Lagardere, L.; Schnieders, M. J.; Piquemal, J. P.; Ren, P.; Ponder, J. W., Tinker 8: Software Tools for Molecular Design. *J Chem Theory Comput* **2018,** *14* (10), 5273-5289.
199. Åqvist, J.; Wennerström, P.; Nervall, M.; Bjelic, S.; Brandsdal, B. O., Molecular dynamics simulations of water and biomolecules with a Monte Carlo constant pressure algorithm. *Chemical physics letters* **2004,** *384* (4-6), 288-294.
200. Allen, M. P.; Tildesley, D. J., *Computer simulation of liquids*. Oxford university press: 2017.
201. Nosé, S.; Klein, M., Constant pressure molecular dynamics for molecular systems. *Molecular Physics* **1983,** *50* (5), 1055-1076.
202. Toukmaji, A.; Paul, D.; John Jr, A. In *Distributed Patrticle-Mesh Ewald: A Parallel*

Ewald Summation Method, PDPTA, 1996; pp 33-43.
203. Louwerse, M. J.; Baerends, E. J., Calculation of pressure in case of periodic boundary conditions. *Chemical physics letters* **2006,** *421* (1-3), 138-141.
204. Tuckerman, M.; Berne, B. J.; Martyna, G. J., Reversible multiple time scale molecular dynamics. *The Journal of chemical physics* **1992,** *97* (3), 1990-2001.
205. Liu, C.; Piquemal, J.-P.; Ren, P., AMOEBA+ Classical Potential for Modeling Molecular Interactions. *Journal of Chemical Theory and Computation* **2019**.
206. Aviat, F.; Lagardère, L.; Piquemal, J.-P., The truncated conjugate gradient (TCG), a non-iterative/fixed-cost strategy for computing polarization in molecular dynamics: Fast evaluation of analytical forces. *The Journal of chemical physics* **2017,** *147* (16), 161724.
207. Lagardère, L.; Aviat, F. l.; Piquemal, J.-P., Pushing the Limits of Multiple-Time-Step Strategies for Polarizable Point Dipole Molecular Dynamics. *The journal of physical chemistry letters* **2019,** *10*, 2593-2599.